JAPANESE PATENT OFFICE

PATENT ABSTRACTS OF JAPAN

(11)Publication number: 11341076
(43)Date of publication of application: 10.12.1999

(51)Int.Cl.
H04L 12/66
G06F 17/30
H04L 12/56

(21)Application number: 10317235
(22)Date of filing: 09.11.1998

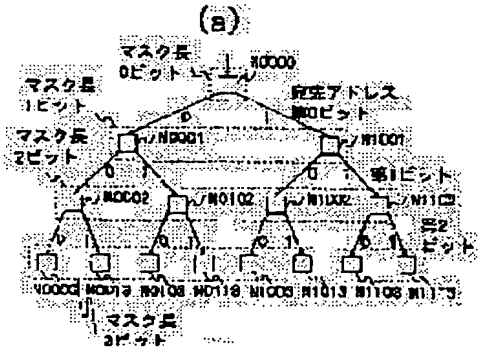
(71)Applicant: HITACHI LTD
(72)Inventor: SUKAI KAZUO
AIMOTO TAKESHI
MATSUYAMA NOBUHITO
AKAHA SHINICHI
SAKO YOSHITO
TANABE NOBORU

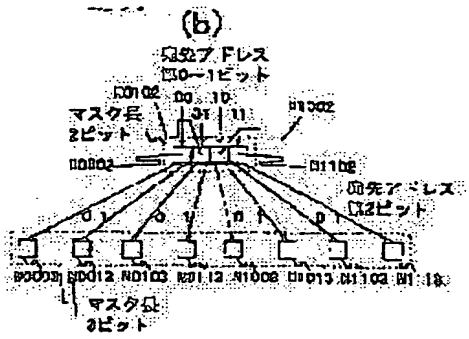
(30)Priority
Priority number: 11999999 Priority date: 23.03.1998 Priority country: JP

(54) NETWORK REPEATER AND NEXT NETWORK TRANSFER DESTINATION RETRIEVAL METHOD

(57)Abstract:

PROBLEM TO BE SOLVED: To eliminate a time for retrieval processing by a host number bit of a sub-network address by extending a node by the host number bit of the sub-network address at the time of retrieval of a host bit correspondence part of the sub-network address.
SOLUTION: When nodes N0002, N0102, N1002 and N1102 of mask length 2 bit are extended at a fixed





position of a memory, an address where the mask length 2 bit nodes are extended from values of a 0 and first bits are obtained, any of the mask length 2 bit nodes is jumped to, and a time is reduced by a time for node retrieval done twice. Generally speaking, when a node of a mask length (m) bit is extended at a fixed position on the memory and jumped to at a time, it is possible to reduce the time for retrieval by a time spent for crossing a node for (m) times of the node in total from a mask length 0 bit. On the other hand, as memory efficiency deterioration is invited by extending the node of the mask length (m) bit of the m-powered of 2 number regardless whether or not there is the node, a value (m) is decided from a trade-off between the memory efficiency and performance.

LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

E5388

(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号

特開平11－341076

(43)公開日　平成11年(1999)12月10日

(51)Int.Cl.[6]	識別記号	ＦＩ		
Ｈ０４Ｌ　12/66		Ｈ０４Ｌ	11/20	Ｂ
Ｇ０６Ｆ　17/30		Ｇ０６Ｆ	15/411	
Ｈ０４Ｌ　12/56		Ｈ０４Ｌ	11/20	１０２Ｄ

審査請求　未請求　請求項の数11　ＯＬ　（全 25 頁）

(21)出願番号　特願平10－317235

(22)出願日　平成10年(1998)11月９日

(31)優先権主張番号　特願平11－999999

(32)優先日　平10(1998)３月23日

(33)優先権主張国　日本（ＪＰ）

(71)出願人　000005108
株式会社日立製作所
東京都千代田区神田駿河台四丁目６番地

(72)発明者　須貝　和雄
神奈川県秦野市堀山下１番地　株式会社日立製作所汎用コンピュータ事業部内

(72)発明者　相本　毅
東京都国分寺市東恋ヶ窪一丁目280番地
株式会社日立製作所中央研究所内

(72)発明者　松山　信仁
神奈川県秦野市堀山下１番地　株式会社日立インフォメーションテクノロジー内

(74)代理人　弁理士　小川　勝男

最終頁に続く

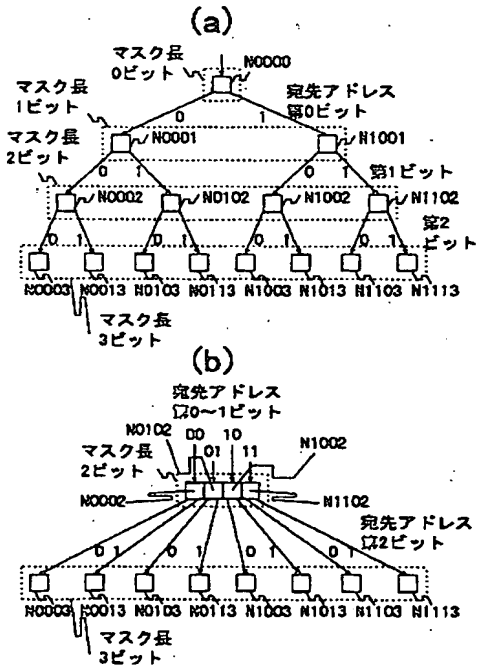
(54)【発明の名称】　ネットワーク中継装置及びネットワーク次転送先検索方法

(57)【要約】

【課題】ルータ等のネットワーク中継装置において、パケットを転送するための転送際の経路検索処理を高速に実行する。

【解決手段】経路情報に基づいてネットワーク中継装置に入ってきたパケットの宛先アドレスからパケットの転送先アドレスを検索する際に、経路検索のためのデータ構造を、受け取ったパケットの宛先アドレスの上位ビットから１ビットずつ検査してゆく２分木検索のｐ（ｐは２以上の整数）段分を一つの２のｐ乗分木にし、２分木のｐ数段の検索を１段で行うにより、高速に経路を検索する。

図9



【特許請求の範囲】

【請求項１】複数のネットワークを接続するネットワーク中継装置であって、前記ネットワークの一つを接続するポートと、前記ポートに接続され、該ポートに接続されたネットワークとのインタフェースを制御するネットワークインタフェース部と、前記ネットワークインタフェース部と装置内通信路を介して接続され、前記ネットワークインタフェースから受け取ったパケットのルーティング処理を行うルーティング処理部とを有し、
前記ルーティング処理部は、経路情報保持手段と、前記経路情報保持手段に保持された経路情報に基づいて前記受け取ったパケットの次に転送すべき経路を算出する次経路検索手段とを包含し、
前記次経路検索手段は、前記受け取ったパケットの宛先アドレスの上位ビットから１ビットずつ検査してゆく２分木検索のｐ（ｐは２以上の整数）段分を、一つの２のｐ乗分木にし、２分木のｐ数段の検索を１段で行うことを特徴とするネットワーク中継装置。

【請求項２】前記次経路検索手段は、一つの２分木ノードと、その直下につながるｐ－１段分の合計２のｐ乗－１個分の２分木ノードを一つの前記２のｐ乗分木ノードにまとめ、まとめられる最下段の２の（ｐ－１）乗個の２分木ノードに、それより上段のノードに割り付けられていた経路データを埋め込み、前記２のｐ乗分木ノードを２分木を２の（ｐ－１）乗個分併せた形で構成することを特徴とする請求項１記載のネットワーク中継装置。

【請求項３】前記次経路検索手段は、２分木を複数個併せるときに、一つだけ持てば良い要素を一つだけ持つようにすることを特徴とする請求項２記載のネットワーク中継装置。

【請求項４】前記次経路検索手段は、２のｐ乗分木ノードを検索のために読むときにノード全てを読まずに、２のｐ乗分木ノードを作成するときに併せた２の（ｐ－１）乗個の２分木ノードの内の、いずれか一つに対応するデータのみを読むことを特徴とする請求項２記載のネットワーク中継装置。

【請求項５】前記次経路検索手段は、各ノードにそのノード自身のマスク長を格納せずにそのノードの直ぐ下に繋がるノードのマスク長を格納することにより、ノードのマスク長を、そのノードのデータを読む前に知り、宛先アドレスの、ノードのマスク長で示されるビット位置から、そのビット位置＋ｐ－１までの値に従い、ノードのデータの内の読み込む部分を選択することを特徴とする請求項２記載のネットワーク中継装置。

【請求項６】前記次経路検索手段は、各ノードの最初に読み込むデータ内に、そのノードに経路が割り付けられているか否かを示すフラグを設け、最初に、このフラグを読み込み、経路が割り付けられていないノードでは経路情報を読み込まないことを特徴とする請求項４記載のネットワーク中継装置。

【請求項７】前記ネットワーク中継装置はルータであることを特徴とする請求項１記載のネットワーク中継装置。

【請求項８】複数のネットワークを接続するネットワーク中継装置であって、前記ネットワークの一つを接続するポートと、前記ポートに接続され、該ポートに接続されたネットワークとのインタフェースを制御するネットワークインタフェース部と、前記ネットワークインタフェース部と装置内通信路を介して接続され、前記ネットワークインタフェースから受け取ったパケットのルーティング処理を行うルーティング処理部とを有し、
前記ルーティング処理部は、経路情報保持手段と、前記経路情報保持手段に保持された経路情報に基づいて前記受け取ったパケットの次に転送すべき経路を算出する次経路検索手段とを包含し、
前記次経路検索手段は、次経路の検索を宛先アドレスの上位ビットから１ビットずつ検査してゆく２分木検索により行い、検査を行うビット位置を、マスク長に対応させることによりマスク付きの一致検索を行い、マスク長ｍ（ｍは自然数）ビットのノードを、２のｍ乗個、記憶手段上の決まった位置に展開し、それぞれのマスク長ｍビットのノードを、それぞれ、宛先アドレスの第０ビットから第ｍ－１ビットまでが取りうる値に１対１に対応させ、宛先アドレスの第０ビットから第ｍ－１ビットの値に従い、マスク長ｍビットのノードの一つを選択することを特徴とするネットワーク中継装置。

【請求項９】複数のネットワークを接続するネットワーク中継装置であって、前記ネットワークの一つを接続するポートと、前記ポートに接続され、該ポートに接続されたネットワークとのインタフェースを制御するネットワークインタフェース部と、前記ネットワークインタフェース部と装置内通信路を介して接続され、前記ネットワークインタフェースから受け取ったパケットのルーティング処理を行うルーティング処理部とを有し、
前記ルーティング処理部は、経路情報保持手段と、前記経路情報保持手段に保持された経路情報に基づいて前記受け取ったパケットの次に転送すべき経路を算出する次経路検索手段とを包含し、
前記次経路検索手段は、次経路の検索を宛先アドレスの上位ビットから１ビットずつ検査してゆく２分木検索により行い、マスク長０ビットからｋビットまでの２分木ノードを、先頭からのビット数が所定数の部分を前記次経路検索手段の内蔵記憶手段内に置き、マスク長ｋ＋１ビット以降の２分木ノードを検索手段の外部記憶手段内に置き、第０から第ｋビットまでの検索処理と、第ｋ＋１ビット以降の検索処理をパイプライン処理することを特徴とするネットワーク中継装置。

【請求項１０】複数のネットワークを接続し、前記ネットワークの一つから受け取ったパケットを経路情報に基づいて次の転送先に送出するネットワーク中継装置にお

けるネットワーク次転送先検索方法であって、
前記受け取ったパケットの宛先アドレスの上位ビットから１ビットずつ検査してゆく２分木検索のｐ（ｐは２以上の整数）段分を、一つの２のｐ乗分木にし、２分木のｐ数段の検索を１段で行うことを特徴とするネットワーク次転送先検索方法。
【請求項１１】前記ルーティング処理部内次経路検索手段で行う次経路検索処理をハードウエアで行うことにより、ノード内のデータを読み出すアドレスの計算処理と経路情報の読み込み処理を並列に行うことを特徴とする請求項１乃至請求項９のいずれかに記載のネットワーク中継装置。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】本発明は、コンピュータネットワークシステムにおけるルータ等のネットワーク中継装置に関し、特にネットワーク中継装置に入ってきたパケットの宛先アドレスからパケットの転送先アドレスを高速に検索するのに適したネットワーク中継装置及びネットワーク次転送先検索方法に関する。
【０００２】
【従来の技術】ネットワークシステムにおいては、複数のサブネットを接続するためにルータ等のネットワーク中継装置が用いられる。ルータは接続されているサブネットから受け取ったパケットの宛先アドレスを調べてパケットの転送先を決定し、転送先のルータやホストが接続されたサブネットに受け取ったパケットを転送する。図１は複数のサブネットがルータによって接続された一般的なネットワークシステムの構成を示す。
【０００３】図１において、Ｒ１及びＲ２はルータ、ＳＮ１はルータＲ１のポートＰ１１に接続されたサブネットワーク、ＳＮ２はルータＲ１のポートＰ１２及びルータＲ２のポートＰ２１に接続されたサブネットワーク、Ｈ１０及びＨ１１はサブネットＳＮ１に接続されたホスト、Ｈ２０及びＨ２１はサブネットＳＮ２に接続されたホスト、Ｈ３０及びＨ３１はサブネットＳＮ３に接続されたホストである。
【０００４】ホストＨ１０からホストＨ２１にパケットを送る場合、ルータＲ１はパケット内にヘッダ情報として格納されている宛先アドレスＤＡを調べて、宛先のホストＨ２１がサブネットＳＮ２上にあり、かつサブネットＳＮ２がルータＲ１に直接接続されていることを認識する。そして、ルータＲ１はパケットをサブネットＳＮ２に接続されているポートＰ１２に出力し、出力するときに次に転送するアドレス（次ホップアドレス）を宛先ホスト自身（Ｈ２１）とする。
【０００５】また、ホストＨ１０からホストＨ３１にパケットを送る場合、ルータＲ１はパケット内にヘッダ情報として格納されている宛先アドレスＤＡを見て、宛先のホストＨ３１がサブネットＳＮ３上にあり、かつサブネットＳＮ３がルータＲ１に直接には接続されておらず、ルータＲ２経由で接続されていることを認識する。ルータＲ１はパケットをルータＲ２が接続されているサブネットＳＮ２に接続されているポートＰ２１に出力し、出力するときに次に転送するアドレス（次ホップアドレス）をルータＲ２とする。この場合、ルータＲ２はパケットを受け取ると、ルータＲ１と同様に、宛先アドレスＤＡを見てパケットをホストＨ３１に転送する。
【０００６】次に、ルータがパケットを受け取ったときに、次に転送するアドレス、及びパケットを出力するポートを検索するときの検索仕様を図２を使用し説明する。ＴＢＬは経路検索テーブルであり、このテーブルは人手で入力された構成定義情報、及びルータ間での接続情報のやりとりにより得られた情報から作成される。
【０００７】経路検索テーブルＴＢＬは、サブネットワークアドレスとサブネットワークマスク長の組を検索のキーとして、出力ポート、次ホップアドレス、及びサブネットワークが直接接続されているか否かの情報（以後、次ホップ情報と呼ぶ）を検索するものである。
【０００８】経路検索仕様においては、最上位ビット側からサブネットワークマスク長のビット数だけを有効とするマスクを宛先アドレスに掛けたものをサブネットワークアドレスと比較する。比較の結果、一般的にはマスク長の異なる複数のエントリＥ１、Ｅ２、Ｅ４が一致し、一致したエントリの内でマスク長が最長のものＥ２の次ホップ情報（次ホップ２）を検索結果とする。
【０００９】この検索仕様に従った検索を高速に行う方法として、Ｒａｄｉｓｈアルゴリズムがある。Ｒａｄｉｓｈアルゴリズムについては例えば、ＵＮＩＸ　ＭＡＧＡＺＩＮＥ　１９９７．４　ｐｐ．２０－２５　山口英「カーネルを読もう（８）ＩＰ層における経路制御機構（２）」に解説されている。
【００１０】
【発明が解決しようとする課題】上記Ｒａｄｉｓｈアルゴリズムは、左右にポインタを持つ複数の頂点（ノード）をポインタでつないだ木から構成される木構造の各ノードに経路エントリをマップし、この木を辿るときには、各ノードの左右のどちらかのポインタを辿り次のノードに移動することにより、目的の経路エントリがマップされたノードにたどり着くアルゴリズムである。
【００１１】まず、図３を参照して木の構造を説明する。考え方はビット長には依存しないので、図３では理解し易いようアドレス長を３ビットとして説明する。
【００１２】図３に示すように、各ノードを、木の上から順にマスク長０ビット、１ビット、２ビット、３ビットのノードと呼ぶ。
【００１３】マスク長０ビットのノードＮ０００００では宛先アドレスの第０ビットが０か１かに従い左／右のポインタを辿ることによりマスク長１ビットのノードＮ０００１、Ｎ１００１に移り、マスク長１ビットのノード

では第１ビットが０か１かに従い左／右のポインタを辿ることによりマスク長２ビットのノードＮ０００２、Ｎ０１０２、Ｎ１００２、Ｎ１１０２に移り、マスク長２ビットのノードでは第２ビットが０か１かに従い左／右のポインタを辿ることによりマスク長３ビットのノードＮ０００３、Ｎ００１３、Ｎ０１０３、Ｎ０１１３、Ｎ１００３、Ｎ１０１３、Ｎ１１０３、Ｎ１１１３に移る。

【００１４】検索したい宛先アドレスについて、この木のマスク長０ビットのノードＮ００００から順に各ビットが０か１かに従いポインタを辿った場合、マスク長０ビットのノードは宛先アドレスがどの場合にも通過し、マスク長１ビットのノードＮ０００１、Ｎ１００１は左から順に宛先アドレスの各ビットが０ＸＸ、１ＸＸの場合に通過し、マスク長２ビットのノードＮ０００２、Ｎ０１０２、Ｎ１００２、Ｎ１１０２は左から順に宛先アドレスの各ビットが００Ｘ、０１Ｘ、１０Ｘ、１１Ｘの場合に通過し、マスク長３ビットのノードＮ０００３、Ｎ００１３、Ｎ０１０３、Ｎ０１１３、Ｎ１００３、Ｎ１０１３、Ｎ１１０３、Ｎ１１１３は左から順に宛先アドレスの各ビットが０００、００１、０１０、０１１、１００、１０１、１１０、１１１の場合に通過する。ここで、Ｘはそのビット値が０または１のどちらでも良いことを示す。

【００１５】したがって、マスク長０ビットのノードＮ００００は、宛先アドレスがサブネットワークアドレス０００／０に属する場合に通過し、マスク長１ビットのノードＮ０００１、Ｎ１００１は、宛先アドレスがサブネットワークアドレス０００／１、１００／１に属する場合に通過し、マスク長２ビットのノードＮ０００２、Ｎ０１０２、Ｎ１００２、Ｎ１１０２は、宛先アドレスがサブネットワークアドレス０００／２、０１０／２、１００／２、１１０／２に属する場合に通過し、マスク長３ビットのノードＮ０００３、Ｎ００１３、Ｎ０１０３、Ｎ０１１３、Ｎ１００３、Ｎ１０１３、Ｎ１１０３、Ｎ１１１３は、宛先アドレスがサブネットワークアドレス０００／３、００１／３、…、１１１／３に属する場合に通過する。ここで、表記法“ｓｓｓ／ｍ”の“ｓｓｓ”はサブネットワークアドレス、ｍはマスク長を表すものとする。

【００１６】上記の通り、この木の各ノードは、サブネットワークアドレスとマスク長が異なる全サブネットに１対１に対応している。

【００１７】そこで、図４に示す経路テーブルエントリに対応するノードＮ００００、Ｎ００１３、Ｎ０１０２、Ｎ１００１、及びＮ１１０３に“＊”を付け、検索したい宛先アドレスＤＡ０１１を、この木の上から各ビットが０か１かに従いポインタを辿ったときに通過する“＊”を付けたノードＮ００００、Ｎ０１０２が、マスク付きの検索で一致するエントリに対応することが分かる。そこで、経路テーブルエントリが複数一致した場合は最もマスク長が長いサブネットワークを選択する、という規則に対応し、一致した“＊”付きノードＮ００００、Ｎ０１０２の内、最も末端に近いノードＮ０１０２に割り付けられた経路情報を経路テーブルの検索結果とする。

【００１８】上記検索方法から分かるように、“＊”が付いておらず、かつ“＊”付きのノードにたどり着くための途中経路にもなっていないノードＮ０００３、Ｎ０１０３、Ｎ０１１３、Ｎ１００３、Ｎ１０１３、Ｎ１１１３、及びＮ１００２は、木から取り除いても検索結果には影響しない。むしろ、最下のノードに“＊”が付いていないときは、最下まで移動せずに検索が終了するために効率的である。そこで、“＊”が付いておらず、かつ“＊”付きのノードにたどり着くための途中経路にもなっていないノードを木から取り除くと図５のようになる。

【００１９】この方法で、アドレス長が３２ビットで、図６に示す経路テーブルのに対応する２分木を描くと、図７のようになり、分岐も“＊”も無いノードの長い列ＮＳ１ができる。このように、左右の片方のポインタだけに次のノードがつながり、かつ経路エントリがマップされていないノードを取り除くことによる高速化法について説明する。

【００２０】この高速化法では、分岐も“＊”も無いノード列ＮＳ１を取り除き、直ぐ上のノードＮ０００００００００の分岐方向（図７では右側）に、取り除かれたノード列ＮＳ１の直ぐ下のノードＮ８５０４０００１５を付ける。その結果、図８に示す形となる。このように途中のノード列を取り除くことを、以後、木の縮退と呼ぶ。

【００２１】次に縮退した木での経路の検索法を説明する。

【００２２】図８に示す例では、マスク長０ビットのノードＮ０００００００００で第０ビットの検索を行った後、マスク長１５ビットのノードＮ８５０４０００１５に跳ぶので、マスク長１５ビットのノードＮ８５０４０００１５で第１５ビットだけを検査したのでは、途中のビット、即ち第１ビットから第１４ビットが検査できない。そこで、第１ビットから第１５ビットの検索を一回の処理で行う為に、宛先アドレスの第１から第１５ビットとノードＮ８５０４０００１５のサブネットワークアドレス０ｘ８５．０４．００．００の第１から第１５ビットの一致比較を行う。比較結果が一致すれば正しいノードにたどり着いたこと、即ち、縮退しない木で１ビットずつ比較してもこのノードにたどり着いたことを意味し、一致しなければ正しくないノードにたどり着いたこと、即ち縮退しない木では行き先が無いことを意味する。

【００２３】ここで、図８に示す例では、第０ビットは

既にテストされ、第０ビットが等しくなる方の分岐が選択されている為、常に一致する。一般に、あるノードにたどり着く毎に正しいノードにたどり着いたか否かを検査していれば、第０ビットからそのノードのマスク長までのビットは宛先アドレスとノードのサブネットワークアドレスとで等しいことが保証されているので、次のノードにたどり着いたときに、前にどのビットまでテストしたかに関らず、第０ビットからノードのマスク長までのビットが宛先アドレスとノードのサブネットワークアドレスとで等しいか否かを調べて良い。

【００２４】このように、Ｒａｄｉｓｈアルゴリズムでは、経路を検索するために宛先アドレスを上から１ビットずつ検査しており、経路検索処理に時間が掛かる、という問題があった。

【００２５】本発明の目的は、パケットを転送するための転送際の経路検索処理を高速に実行するネットワーク中継装置、特にルータを提供することにある。

【００２６】本発明の他の目的は、ルータ等のネットワーク中継装置において、受信したパケットの宛先アドレスからパケットの転送先アドレスを高速に検索するネットワーク次転送先検索方法を提供することにある。

【００２７】

【課題を解決するための手段】上記目的を達成するため、本発明においては、サブネットワークアドレスの上位ビットに対応する部分の検索について、サブネットワークアドレスの上位数ビット分、ノードをメモリ上の決まった位置に展開することにより、サブネットワークアドレスの上位数ビット分の検索処理時間を無くすようにする。

【００２８】また、検索木の、上位数ビット分を検索処理を行うＬＳＩに内蔵し、ＬＳＩ内部のメモリと外部のメモリとの間で検索処理をパイプライン処理することにより、上位数ビット分の検索処理時間を隠すようにするまた、検索木を構成する各ノードを、従来技術での２分木ノードから４分木、８分木、あるいはそれ以上と枝別れの数を２のべき乗で増やし、一つのノードで１ビットでなく、連続する２ビット、３ビット、あるいはそれ以上のビット数を同時に検査し、検索終了までに辿るノードの数を減らすようにする。

【００２９】また、検索木を表現するデータ構造を記憶するためのメモリ量を減らすために、４分木、８分木、あるいは一般に２のｐ乗分木を構成するときに、一つの２分木ノードと、その直下につながるｐ－１段分の合計２のｐ乗－１個分の２分木ノードを一つの２のｐ乗分木ノードにまとめ、まとめられる最下段の２の（ｐ－１）乗個の２分木ノードに、それより上段のノードに割り付けられていた経路データを埋め込むことにより、２のｐ乗分木ノードを２分木を２の（ｐ－１）乗個分併せた形で構成するようにし、さらに、２分木を複数個併せるときに、一つだけ持てば良い要素を一つだけ持つようにする。

【００３０】また、この２分木を複数個併せた形で構成した４分木、８分木、あるいはそれ以上の枝別れ数のノードを、検索のために読むときにノード全てを読むのではなく、必要な部分のみを読むようにし、ノードが大きくなることによるデータの読み込み時間の増大を防ぐ。ノードのデータの内、必要な部分のみを選択するため、ノードのマスク長をノードのデータを読む前に知る必要があり、各ノードには、そのノードのすぐ下につながるノードのマスク長を格納するようにする。また、各ノードの先頭に、そのノードに経路が割り付けられているか否かを示すフラグを設け、最初に、このフラグを読み込み、経路が割り付けられていないノードでは、経路情報を読み込まないようにすることにより、データの読み込み時間の短縮を図る。

【００３１】また、木構造の検索処理を専用ハードウエアで行うことにより、次のノードの読み込む位置のアドレス計算と、経路情報の候補の更新処理を、並列化することにより高速化する。

【００３２】

【発明の実施の形態】本発明をより詳細に説明するため、添付の図面を参照して本発明を実施するための最良の形態を説明する。

【００３３】最初に、本発明が適用されるルータ装置の代表的な構成を図３８を参照して説明する。図３８において、１００はルータ装置、１１０はルーティング制御部、１２０はルータバス、１３０はネットワークインタフェース部、１４０はポート、１５０はサブネットワークである。

【００３４】ネットワークインタフェース部１３０は、ポート１４０に接続されたサブネットワークからパケットを受け取り、受け取ったパケットをルータバス１２０経由でルーティング制御部１１０に送信する。ルーティング制御部１１０はルーティング情報を保持するルーティングテーブルを備え、このルーティング情報を用いて受け取ったパケットの宛先から転送先のサブネットワーク１５０を決定し、当該サブネットワーク１５０が接続されるポート１４０のネットワークインタフェース部１３０にパケットを送信する。ルーティング制御部１１０からパケットを受け取ったネットワークインタフェース部１３０はそのパケットを転送先のサブネットワーク１５０に送出する。なお、ルーティング制御部１１０は、受け取ったパケットのヘッダ情報に基づいてルーティングテーブルに保持するルーティング情報を更新・保守するとともに、ルータ装置１００全体の管理機能を備えている。

【００３５】図３９は、ルータ装置の他の構成例を示すブロック図である。図３９において、２００はルータ装置、２１０はルーティングプロセッサ（ＲＰ）、２２０はルータ装置内通信手段、２３０はネットワークインタ

フェース部、２４０はポート、２５０はサブネットワーク、２６０はルータ管理部である。本構成の場合、図３８に示した構成のルーティング制御部１１０がルーティング機能を実行するルーティングプロセッサ２１０及びルータ装置２００の管理を行うルーティング管理部２６０に分かれるとともに、図３８に示した構成に相当するネットワークインタフェース部２３０及びルーティングプロセッサ２１０から構成される部分を複数備えている。ルーティング管理部２６０は、ルータ装置２００全体の管理機能を備えるとともに、各ルーティングプロセッサ２１０にルーティング情報を配付する。ルータ装置内通信手段２２０は、クロスバスイッチあるいはバス等であり、ルーティングプロセッサ２１０相互の通信やルーティングプロセッサ２１０とルーティング管理部２６０との間の通信を行う。ルーティングプロセッサ２１０は、図３８のルーティング制御部１１０と同様に自分に接続されたネットワークインタフェース部２３０の間のパケット転送を行うとともに、他のルーティングプロセッサ２１０に接続されたサブネットワーク２５０にパケットを転送する場合は、ルータ装置内通信手段２２０を介して該当するルーティングプロセッサ２１０にパケットを転送する。

【００３６】次に、ルーティング制御部１１０及びルーティングプロセッサ２１０において実行される次転送先経路検索処理について説明する。最初に、Ｒａｄｉｓｈアルゴリズムの本発明による高速化法を示す高速化の１番目の方法を図９を参照して説明する。従来のＲａｄｉｓｈアルゴリズムでは、木が縮退していない場合、マスク長０ビットのノードから順に１ビットすつ検索してゆくが、本発明では図９に示すように、マスク長ｍビットのノードをノードが有る場合も無い場合も、全てメモリ上の決まった位置に展開するものである。図９はマスク長２ビットのノードＮ０００２、Ｎ０１０２、Ｎ１００２、Ｎ１１０２をメモリの決まった位置に展開した場合の例である。

【００３７】この場合、従来はまずマスク長０ビットのノードＮ０００００に跳び、第０ビットの値に従い、マスク長１ビットのノードＮ０００１、Ｎ１００１のどちらかに跳び、第１ビットの値に従い、マスク長２ビットのノードＮ０００２、Ｎ０１０２のどちらか、あるいはＮ１００２、Ｎ１１０２のどちらかにたどり着いていた。

【００３８】本発明においては、第０ビット及び第１ビットの値からマスク長２ビットのノードＮ０００２、Ｎ０１０２、Ｎ１００２、Ｎ１１０２を展開してあるアドレスを求め、直接マスク長２ビットのノードＮ０００２、Ｎ０１０２、Ｎ１００２、Ｎ１１０２のいずれかに跳ぶ。これにより、２回のノード検索の時間分、検索時間が短縮される。

【００３９】一般に、マスク長ｍビットのノードをメモリ上に決まった位置に展開し、１回で跳んだ場合、マスク長０ビットからｍ－１ビットまでのノードの計ｍ回のノードを渡る時間分、検索時間が短縮される。一方で、２のｍ乗個のマスク長ｍビットのノードをノードが有る場合も無い場合もメモリ上に展開する必要があるため、メモリ効率が悪くなる。したがって、メモリ効率と性能のトレードオフからｍの値を決めるようにする

次に高速化の２番目の方法を図１０乃至図１２を参照して説明する。図１０に示すようにマスク長ｋビットまでのノードを経路検索を行うＬＳＩ　Ｌ１の内蔵メモリＭＬ１に展開し、マスク長ｋ＋１ビット以降のノードをＬＳＩ　Ｌ１外部のメモリＭ１に展開する。このようにすることにより、ＬＳＩ　Ｌ１内部のメモリＭＬ１のアクセスが高速であること、及びＬＳＩ　Ｌ１内部のメモリＭＬ１と外部のメモリＭ１とが独立したメモリであることから、パイプライン処理できることを利用し、高速化を図るものである。一般的に、ＬＳＩ　Ｌ１に内蔵できるメモリ量は外部に持つことができるメモリ量に比べ少ないが、マスク長が短い内は、ノードの数が少ないという性質があるので、マスク長が短い方のノードをＬＳＩ　Ｌ１に内蔵することができる。

【００４０】図１１にパイプライン処理を行っていない従来の場合のタイムチャートを、図１２に本願発明によるＬＳＩ　Ｌ１内部のメモリＭＬ１と外部のメモリＭ１との間で経路検索処理のパイプライン処理を行っているときのタイムチャートを各々示す。従来は図１１に示すように、あるパケット（パケット１、２、３）の経路検索を行う場合に、パケット１のマスク長ｋビットまでのノードの検索処理ＰＲ１０と、マスク長ｋ＋１ビット以降のノードの検索処理ＰＲ１１、パケット２の同上の処理ＰＲ２０、ＰＲ２１、パケット３の同上の処理ＰＲ３０、ＰＲ３１を順番に行っていた。これに対して、本発明では図１２に示すように、ＬＳＩ　Ｌ１内部のメモリＭＬ１を使用してパケット１のマスク長ｋビットまでのノードの検索処理ＰＲ１０を行った後、ＬＳＩ　Ｌ１外部のメモリＭ１を使用しパケット１のマスク長ｋ＋１ビット以降のノードの検索処理ＰＲ１１を始めると同時に、ＬＳＩ　Ｌ１内部のメモリＭＬ１を使用しパケット２のマスク長ｋビットまでのノードの検索処理ＰＲ２０を始めるようにする。その後、処理ＰＲ２０及びＰＲ１１が終わったら、ＬＳＩ　Ｌ１外部のメモリＭ１を使用しパケット２のマスク長ｋ＋１ビット以降のノードの検索処理ＰＲ２１を始めると同時に、ＬＳＩ　Ｌ１内部のメモリＭＬ１を使用しパケット３のマスク長ｋビットまでのノードの検索処理ＰＲ２０を始める。以後、同様にパケットの検索処理をパイプライン処理で行う。

【００４１】高速化の３番目の方法は、従来一つのノードに２つの分岐先があり１ビットずつ検索していたものを、一つのノードに２のｐ乗の分岐先を設け、同時にｐビットずつ検索することにより、従来に比べ検索時間を１／ｐに短縮するものである。以後、一つのノードに２

のｐ乗の分岐先があるノードのことを２のｐ乗分木ノードと呼ぶ。

【００４２】２のｐ乗分木ノードは、従来の方法である２分木ノードから構成される木を変形することにより作成する。木の変形の方法は、一つの第ｎビット目の２分木ノードと、この２分木ノードの下の第ｎ＋１からｎ＋ｐ－１ビット目の２分木ノードを、一つの２のｐ乗分木ノードに対応させるように行う。例として、２分木から８分木への変形法を図１３乃至図１５に示す。

【００４３】８分木の場合、一つの８分木に対応させる２分木ノードのビット位置のとり方として、以下の３通りがある。

【００４４】（ａ）図１３に示すように、マスク長０～２、３～５、６～８、９～１１、１２～１４、１５～１７、１８～２０、２１～２３、２４～２６、２７～２９、３０～３２ビットのノードを、それぞれ一つの８分木ノードとする場合。

【００４５】（ｂ）図１４に示すように、第１～３、４～６、７～９、１０～１２、１３～１５、１６～１８、１９～２１、２２～２４、２５～２７、２８～３０、３１～３２ビットのノードを、それぞれ一つの８分木ノードとする場合。

【００４６】（ｃ）図１５に示すように、第２～４、５～７、８～１０、１１～１３、１４～１６、１７～１９、２０～２２、２３～２５、２６～２８、２９～３１、３２ビットのノードを、それぞれ一つの８分木ノードとする場合。

【００４７】どの区切り方でも構成可能だが、木全体では、経路の追加、削除を容易に行えるように、上記３通りのビット位置の区切り方の内の一つを使用する。

【００４８】上記３通りのビット位置の区切り方の内、最初の区切り方以外ではマスク長が０ビットから始まっていないので、最初のビットの検索を別に行う必要がある。この検索には、図９に示したマスク長ｍビットのノードをメモリ上に展開する方法を使用する。図１４、図１５に示した区切り位置に対応する８分木の最初のビット分のノードをメモリ上に展開した木の構成を図１６、図１７に示す。

【００４９】図１６に示す構成では、マスク長１～３ビットのノードＮ８０１３、Ｎ８１１３をメモリ上の決まった位置に並べ、それぞれを第０ビットが０か１かに従い選択する。図１７に示す構成では、マスク長２～４ビットのノードＮ８００２４、Ｎ８０１２４、Ｎ８１０２４、Ｎ８１１２４をメモリ上の決まった位置に並べ、それぞれを第０～１ビットが００か０１か１０か１１かに従い選択する。

【００５０】あるいは、図９に示したマスク長ｍビットのノードをメモリ上に展開する方法と、２のｐ乗分木ノードにして複数のビット数を同時に検索する方法とを組み合わせることも可能である。具体的には、図１３、図１４、図１５に示す各ビットの区切り位置の場合に、それぞれ最初に並べるノード数を１、２、４個ではなく、これらの８倍である８、１６、３２個、８の２乗倍である６４、１２８、２５６個、或いは一般に８のｑ乗倍個にし、最初の１回、２回、或いは一般に２のｐ乗分木ノードのｑ回の検索時間を無くすことも可能である。ｐ＝３、ｑ＝１の場合、即ち８分木ノードの１回のノードの検索時間を無くす場合で、図１３、図１４、図１５に示す３通りの各ビットの区切り位置の場合についてのメモリ上へのノードの展開法を、図１８、図１９、図２０にそれぞれ示す。このようにｐ、ｑの値を大きくすると経路検索時間を短縮することができるが、多くのメモリを必要とするので、ｐ、ｑの値はメモリ効率と性能のトレードオフから決めるようにする。

【００５１】次に、４分木ノード、８分木ノード、１６分木ノード、或いは一般に２のべき乗分木のノードの構成法を図２１を使用し説明する。

【００５２】図２１は４分木で、ある一つの２分木ノードＡ、Ｂ、Ｃ、Ｄ、Ｅと、その直下の２個の２分木ノードＡ０、Ａ１、Ｂ０、Ｂ１、Ｃ０、Ｃ１、Ｄ０、Ｄ１、Ｅ０、Ｅ１の、各々合計３個の２分木ノードをまとめて一つの４分木ノードＮ４０１、Ｎ４０２３、Ｎ４１２３、Ｎ４２２３、Ｎ４３２３にする場合の例であり、合計３個の２分木ノードをつぶして、下の方の２分木ノードだけの大きさにする。つぶし方は、マスク長が異なる複数の経路が一致したら、マスク長の長いほうの経路を採用するという経路検索の仕様に従い、経路検索を行った場合に、２分木の場合と４分木の場合とで、経路検索結果が同じになる、という条件を満たすように行う。

【００５３】４分木の場合の、このノードのつぶし方を図２２～図３０に示す。

【００５４】３つのノードが全部ある場合（図２２）、全ノードに経路情報が割り付けられていたら、下のノードの経路情報＊Ａ０、＊Ａ１を残し、上のノードの経路情報＊Ａは削除する。これは、ノードＡの経路が一致したらノードＡ０かノードＡ１のどちらかの経路が必ず一致するので、マスク長が異なる複数の経路が一致したら、マスク長の長いほうの経路を採用するという経路検索の仕様により、＊Ａが使われることが無いからである。

【００５５】上のノードＡに経路情報＊Ａが割り付けられており、下のノードＡ０、Ａ１の内Ａ１にだけ経路が割り付けられていない場合（図２３）、Ａ１の経路情報に、Ａの経路情報＊Ａを入れる。下のノードＡ０、Ａ１の内Ａ０だけ経路が割り付けられていない場合も同様である。

【００５６】下のノードＡ０、Ａ１の両方に経路が割り付けられていない場合（図２４）には、Ａ０、Ａ１の両方の経路情報に、Ａの経路情報＊Ａを入れる。

【００５７】下のノードＡ１が無い場合（図２５）は当

該ノードを補い、経路情報には上のノードＡの経路情報＊Ａを入れ、ノードＡ１の下にはノードが繋がっていないので、ノードＡ１の下のノードへのポインタにはＮＵＬＬを入れる。下のノードＡ０、Ａ１の内Ａ０だけ経路が割り付けられていない場合も同様である。

【００５８】下のノードＡ０、Ａ１の両方が無い場合（図２６）、両方を補い、両方の経路情報に、Ａの経路情報＊Ａを入れ、両方の下のノードへのポインタにはＮＵＬＬを入れる。

【００５９】上のノードに経路が割り付けられていない場合（図２７）、上のノードを只つぶす。

【００６０】上のノードＡに経路情報＊Ａが割り付けられてなく、下のノードＡ０、Ａ１の内、Ａ１にも経路が割り付けられていない場合（図２８）、４分木にした場合もＡ１の経路情報は無い。下のノードＡ０、Ａ１の内、Ａ０に経路が割り付けられていない場合も同様である。

【００６１】下のノードＡ０、Ａ１の両方に経路が割り付けられていない場合（図２９）、４分木にした場合も両方の経路情報は無い。

【００６２】下のノードＡ０だけしかない場合（図３０）には、下のノードＡ１を補う。下のノードＡ１だけしかない場合も同様である。

【００６３】８分木の場合も同様にして、一つにまとめる７個の２分木ノードをつぶして、一番下の４個のノードだけの大きさにする。上の方の３つの２分木ノードのつぶし方の例を２つ図３１に示す。

【００６４】図３１（ａ）は一つにまとめる７個の２分木ノードが全てあるが、その内のいくつかにしか経路情報が割り付けられていない例である。最下の４つのノードの内、経路情報が割り付けられていないノードＡ０１、Ａ１０には、そのノードの上方につながっているノードの内、経路情報が割り付けられている最も下、即ち最もマスク長が長いノード（それぞれ、Ａ、Ａ１）の経路情報＊Ａ、＊ａ１を入れる。

【００６５】図３１（ｂ）は一つにまとめる７個の２分木ノードの内のいくつかしかノードが存在しない例であり、存在しないノードＡ０１、Ａ１０をまず経路が割り付けられていないノードとして補い、図３１（ａ）と同じ規則で経路情報を入れる。最下の４つのノードＡ００、Ａ０１、Ａ１０、Ａ１１の内、補ったノードＡ０１、Ａ１０の下にはノードが繋がっていないので、下のノードへのポインタにはＮＵＬＬを入れる。

【００６６】一般に２のｐ乗分木の場合も同様にして、一つにまとめる２のｐ乗－１個の２分木ノードをつぶして、一番下の２の（ｐ－１）乗個のノードだけの大きさにする。

【００６７】この方法で１ノードの大きさを小さくすることは、メモリ効率の悪化を防ぐ効果がある。試みに、以下に示す近似の下で木を２分木で作成した場合と２のｐ乗分木で作成した場合のメモリ使用量を計算し、２のｐ乗分木にした場合でも、ｐが小さい場合には、メモリ効率が悪くならないことを示す。

【００６８】近似

１．２分木にした場合、経路は全て末端のノードにのみ割り当てられている。

【００６９】２．木の枝は第３２ｂｉｔまで全てある。

【００７０】３．枝の増え方は、全ｂｉｔで一定とする。

【００７１】４．経路数は約１０ｋ経路とする。

【００７２】この近似の下で構成した木の形を図３２に示す。この近似によると、１ｂｉｔ下る毎の木の広がり方は１０ｋの（１／３２）乗、即ち約１．３３倍である。

【００７３】この構成の木を、２分木ノードで構成した場合と４分木ノードノードで構成した場合のメモリ量の比較を、図３３（ａ）を用いて説明する。４分木ノードを構成する３つの２分木ノードの内、上の一つのノードＮ２に着目すると、直下にはノードが約１．３３個付く。即ち、左右各々０．６７の確率でノードＮ２０、Ｎ２１が付く。したがって、４分木ノードを構成する２分木ノードＮ２、Ｎ２０、Ｎ２１の合計のメモリ量のノードの存在確率を考慮した平均は、２分木ノード（１＋１．３３）個分、即ち２．３３個分である。４分木ノードで木を構成すると、この３つの２分木ノードが一つの４分木ノードＮ４になり、この一つの４分木ノードのメモリ使用量は２分木ノードＮ２０、Ｎ２１の２個分である

同様にして、この構成の木を、２分木ノードで構成した場合と８分木ノードノードで構成した場合のメモリ量の比較を、図３３（ｂ）を用いて説明する。８分木ノードを構成する７つの２分木ノードＮ２、Ｎ２０、Ｎ２１、Ｎ２００、Ｎ２０１、Ｎ２１０、Ｎ２１１の内、最上の一つのノードＮ２に着目すると、直下にはノードＮ２０、Ｎ２１が約１．３３個付き、そのさらに下にはノードＮ２００、Ｎ２０１、Ｎ２１０、Ｎ２１１が約１．３３の２乗、即ち約１．７８個付く。したがって、８分木ノードを構成する２分木ノードＮ２、Ｎ２０、Ｎ２１、Ｎ２００、Ｎ２０１、Ｎ２１０、Ｎ２１１の合計のメモリ量のノードの存在確率を考慮した平均は、２分木ノード（１＋１．３３＋１．７８）個分、即ち４．１１個分である。

【００７４】８分木ノードで木を構成すると、この７個の２分木ノードが一つの８分木ノードＮ８になり、この一つの８分木ノードのメモリ使用量は２分木ノードＮ２００、Ｎ２０１、Ｎ２１０、Ｎ２１１の４個分である。

【００７５】同様にして２分木の場合と４、８、１６、３２、６４、１２８、２５６分木の場合のメモリ容量の比較を行った結果を以下に示す。

【００７６】

2分木:　4分木＝　2．33:　2＝0．86倍
2分木:　8分木＝　4．11:　4＝0．97倍
2分木:　16分木＝　6．48:　8＝1．23倍
2分木:　32分木＝　9．65:　16＝1．66倍
2分木:　64分木＝13．86:　32＝2．31倍
2分木:128分木＝19．49:　64＝3．28倍
2分木:256分木＝26．98:128＝4．74倍

経路数が1M経路の場合には、この近似によると、1ｂｉｔ下る毎の木の広がり方は1Mの（1／32）乗、即ち約1．54倍になり、2分木の場合と4、8、16、32、64、128、256分木の場合のメモリ量の比は、以下の通りとなる。

【0077】

2分木:　4分木＝　2．54:　2＝0．79倍
2分木:　8分木＝　4．91:　4＝0．81倍
2分木:　16分木＝　8．56:　8＝0
2分木:　32分木＝14．19:　16＝1
2分木:　64分木＝22．85:　32＝1
2分木:128分木＝36．18:　64＝1
2分木:256分木＝56．72:128＝2

結論として、上記仮定の下では10k経路時まで、1M経路時には16分木までなら、使用効率は良くなる。256分木にした場合リ使用量は10k経路時に3．28倍、1M2．26倍までしか増えない。このようにメ率があまり悪くならない理由として、以下のられる。

【0078】（1）p段分のノードを纏めるり、纏める前のノードの合計よりも、纏めたがコンパクトになる。即ち、

3個の2分木が、2分木のノードの2倍の大きさの4分木になる。

【0079】

7　〃　4　〃　8　〃
15　〃　8　〃　16　〃
31　〃　16　〃　32　〃
63　〃　32　〃　64　〃
127　〃　64　〃　128　〃
255　〃　128　〃　256　〃

（2）1ｂｉｔ下る毎の木の広がりが大きく、1ノード内のデータ使用効率が良い。（サポート経路数が増える程、木の広がりが大きくなるので、1ノード内のデータ使用効率は良くなる。）

さらに、4、8、16、…分木ノードでは、2分木ノードを2、4、8、…個まとめて扱うので、一つにまとめられる2分木ノード間で一つだけ持てば良い要素は一つだけ持てばよく、これにより4、8、16、…分木ノードのメモリ量をさらに小さくできる。一つにまとめられる2分木ノード間で一つだけ持てば良い要素には、サブネットワークアドレス及びサブネットワークマスク長があるが、サブネットワークマスク長については後述するようにこのノード自身のサブネットワークマスク長ではなく、このノードの直下のノードのサブネットワークマスク長を持つようにするので、メモリ量を小さくする効果は無い。

【0080】2分木ノードの構造、及びこの2分木ノードを2つ併せ、2分木間で一つだけ持てば良いサブネットワークアドレスを一つだけ持つようにした場合の4分木ノードの構造を各々図34、図35に示す。図34は2分木ノードの構造を示す図であり、次のノードのマスク長0、1は、このノード自身のサブネットワークマスク長ではなく、このノードの直下のノードのサブネットワークマスク長である。このように、自分自身でなく直下のノードのマスク長を設定する理由は高速化のためであり、その説明は図36を参照して後述する。Ｆｌａｇ0及びＦｌａｇ1は、このノードに対応するサブネットワークがこのルータに直接つながるか、他のつ以上経由してつながるかを示すビット、及を使用し後述するように、このノードが経路られているノードか否か、即ち、図8に示しはこのノードが“＊”が付いているノードかフラグ、他である。Ｆｌａｇ0とＦｌａｇ1値を入れる。これは、ワードW0とワードWけを読めば良いようにするためである。このードの全てを読むのではなく、一部分を読む高速化については図36を使用し後述する。へのポインタ0、1は、宛先アドレスのこのスク長で示されるビット位置の値が、それそときに次に辿るノードへのポインタである。ワークアドレスは、このノードに対応するネークアドレスである。出力ポート番号及び次レスは、このノードに割り付けられた経路情入ってきたパケットを出力すべきポート及て上のパケットを送るべきルータのアドレスて

【0081】4分木ノードでは、この2分木つ併せ、併せたときに一つだけ持てば良いうだけ持つようにする。一つだけ持てば良いうネットワークアドレスだけである。この方法4分木ノードの構造を図35に示す。

【0082】図34、図35に示す例では、ドは、2のべき乗の大きさである16バイト入りきらない大きさになっているが、4分木し、1ノード内にサブネットワークアドレス

9３倍
１３倍
４０倍
７７倍
２６倍
は８分木
ろメモリ
も、メモ
絡時に
り使用効
項が挙げ

とによ
のノード

ータを一
、図３７
割り付け
木の例で
かを示す
には、同じ
の一つだ
うに、ノ
とによる
のノード
ノードのマ
０、１の
サブネット
ブネットワ
ホップアド
報であり、
そのポート
ある。
ノードを
ータを
ータに
で作

2
分

のべき乗の大きさである
になっている。８分木ノー
レスを１ノードで一つだ
トの大きさに収まった上
の領域は他の情報を入れ
にまとめる２分木の数を
に対し、一つのノードの

ードの大きさを２のべき
／wの構成を非常に簡単
こできる例とその利点を

４分木ノードが３２バ
バンクで構成していた
バンク境界にまたがる
ミックＲＡＭを使用し
領域がＲｏｗアドレス
利点として挙げられ

ノード内の各要素のア
ドへのポインタとその
算でなく、アドレスの
下位ビットをオフセッ
木ノードが３２バイト
る要素のアドレスは、
スの２の５乗ビット以
へのオフセットをア
乗ビットに割り付け
れる。
えば４分木ノードが
に保持する次のノー
ドの先頭のバイトアド
ノード内で１ポインタ
減らせることが利点とし

１６、…分木ノードにし
なり、検索処理時に検索
全て読み込むと、ノード
時間が伸び、性能低下要因
この問題は、ノードを大き
てを読み込まずに一部だけ
避する。この方法につい
る。
この場合の例であり、既に図
こマスク長mビットの４分木
mビット目の値が０の場合に
１の場合に対応する２分木ノ
ることから、宛先アドレスの
対応する方の２分木ノードの部
より、ノードの大きさが大きく

なっても２分木ノードの場合と同じデータ量を読み込むようにする。このとき、図３５で示した、一つにまとめられる２分木ノード間で一つだけ持つ要素であるサブネットワークアドレスは、宛先アドレスのmビット目の値に係わらず読み込むようにする。

【００８９】さらに、宛先アドレスのm＋１ビット目の値を見て、２分木ノードで２つ存在した次ノードへのポインタの内、一方だけを読み込むようにすることにより、読み込むデータ量をさらに少なくする。

【００９０】この方法は２分木の場合でも使用できる。例えばmビット目の２分木の場合には、宛先アドレスのmビット目の値を見て、２つの次ノードへのポインタの内、一方だけを読み込むようにする。

【００９１】上記方法を全て行い、結局、このノードのマスク長をmとした場合、宛先アドレスの第m、m＋１ビットの値が００か、０１か、１０か１１かに従い、それぞれ（Ｗ０→Ｗ４→Ｗ５→Ｗ６）、（Ｗ１→Ｗ４→Ｗ５→Ｗ６）、（Ｗ２→Ｗ４→Ｗ５→Ｗ７）、（Ｗ３→Ｗ４→Ｗ５→Ｗ７）の順にデータを読み込むようにする。

【００９２】このように、あるノードの一部分だけを読み込むためにはこのノードのマスク長mを知る必要があり、このノードのマスク長mは１ノードのデータ読み込みの最初に読み込むか、この情報を一つ前のノードに移して一つ前のノードのデータ読み込み時に読み込む必要がある。ノードのマスク長mを１ノードのデータ読み込みの最初に読み込む方法は、宛先の第mビット目の値の抽出のための検索処理ＬＳＩ内のゲートディレイ、及び、次に読み込む部分のアドレスをメモリに出力してからメモリからのデータを検索処理ＬＳＩ内に読み込むまでの時間であるメモリリードレイテンシだけ、マスク長mを読み込んでから次に読み込む部分を選択して読み込むまで時間が空いてしまうので、ノードの一部だけを読むことによる性能向上効果が少なく、ノードのマスク長mを一つ前のノードに移し、一つ前のノードのデータの読み込み時に読み込む方が性能向上効果がある。

【００９３】さらに、ノードのマスク長mを一つ前のノードに移す場合、１ノードのデータを読み込む順序を、１番目に次のノードのマスク長m、次のノードへのポインタ、次にサブネットワークアドレス、出力ポート番号、及び、次ホップアドレスの順にすることにより、次のノードの最初に読み込む部分のアドレスが最も早く計算できるようにする。

【００９４】次のノードへのポインタは、次のノードのメモリ領域の先頭部分を指しており、次のノードの先頭から最初に読み込む部分までのアドレスのオフセットは、次のノードのマスク長mを読み込み、宛先アドレスの該当ビット位置の値を検査することにより、得られる。

【００９５】次に、１ノード内で、条件によっては、読み込む必要が無い要素を、条件に従い、読み込まないよ

うにすることで、読み込みの時間を削減することで、高速化を図る方法について図３７を使用し説明する。

【００９６】図３７は４分木の場合の例である。Ｒａｄｉｓｈアルゴリズムでは、全ノードに経路が割り付けられているわけではなく、枝の分岐の個所では、経路が割り付けられていなくてもノードを設ける必要がある。図３７に示すように、ノードデータの最初に読み込むフラグ内に、このノードが経路が割り付けられているノードか否かの情報を入れておき、経路が割り付けられていないノードでは、出力ポート、及び、ネクストホップアドレスを読み込まないようにすることで、読み込み時間の短縮が図れる。このノードが経路が割り付けられているノードか否かの情報は、１ビットで表現できるので、この情報を読み込むことによる読み込み時間の増大は小さい。

【００９７】この方法では、このノードのマスク長をｍとすると、宛先アドレスの第m、ｍ＋１ビットの値が００で、Ｗ０を読み、Ｆｌａｇ００から、４分木を構成する０番目の２分木に経路情報が無いと判った場合、Ｗ４だけを読めば良く、経路情報が有ると判った場合にだけ、図３６に示すようにＷ４→Ｗ５→Ｗ６の順に読めば良い。宛先アドレスの第m、ｍ＋１ビットの値が０１、１０、１１の場合も同様である。

【００９８】次にここまでで示した方式の経路検索を行い、経路検索の結果得られた宛先にパケットを転送するルーティングプロセッサ（図３９の２１０）の内部構成例について図４０を用いて説明する。

【００９９】図４０において、ルーティングプロセッサ２１０は、パケット転送処理を行う転送処理部２１１と、転送処理部２１１が、ルーティングプロセッサに入って来たパケットを転送するまでの間、一時的にパケットデータを格納しておくパケットバッファメモリ２１２と、パケットのヘッダ情報に基づき経路検索を行う経路検索処理部２１３と、経路検索処理部２１３が検索する経路テーブルを格納する経路テーブルメモリ２１４を有する。

【０１００】次に、ルーティングプロセッサ２１０におけるパケットの中継処理の動作を説明する。なお、パケット中継処理を行う前に、ルータ管理部（図３９の２６０）が、ルータ装置内通信手段２２０に接続しているルーティングプロセッサ２１０に、それぞれの経路テーブルを配布しており、各ルーティングプロセッサ２１０は経路テーブルメモリ２１４に経路テーブルを格納している状態とする。

【０１０１】転送処理部２１１はパケットバッファメモリ２１２に格納されたパケットデータの内、パケットのヘッダ情報を抽出して経路検索処理部２１３へ渡し、経路検索処理部２１３は、受信パケットのヘッダ内の宛先アドレスを用いて経路テーブルメモリ２１４内に格納されている経路テーブルの検索を行い、検索結果として、経路情報を転送処理部２１１へ渡す。経路情報は、経路が存在するか否かのＦｌａｇ、図３４乃至図３７に示すＦｌａｇ、次転送先アドレス、出力ルーティングプロセッサ（図３９の２１０）の番号、及び、出力ポート（図３９の２４０）の番号である。転送処理部２１１は、経路検索処理部２１３から渡された経路情報に従い、出力先が自ルーティングプロセッサ２１０に繋がるポート（図３９の２４０）の場合、ネットワークインタフェース部（図３９の２３０）にパケットを転送し、他ルーティングプロセッサ（図３９の２１０）に繋がるポート（図３９の２４０）の場合、ルータ装置内通信手段（図３９の２２０）にパケットを転送する。

【０１０２】以上、２分木検索方式（以下方式１と呼ぶ）、２のｐ乗分木検索方式（以下方式２と呼ぶ）、およびマスク長ｍビットのノードを経路テーブルメモリ２１４上に展開する方式（以下方式３と呼ぶ）という各検索方式について説明した。次に、図４０の経路検索処理回路２００が上記の方式を用いて経路検索処理を行う際のフローチャートについて図４１を用いて説明する。図４１のフローチャートでは、方式２と方式３とを組み合わせた場合の例を示す。この例では、経路アドレスの第０ビットから第（ｍ－１）ビットの値に従って、経路テーブルメモリ２１４上の決まった位置に展開されたノードの一つを選択する。以下ではこのように選択し、検索の最初に読込むノードを初段ノードと呼ぶ。第ｍビット以降は経路アドレスをｐビットずつ検索し、２のｐ乗分木を検索する。図４０の経路テーブルメモリ２１４には、上記の方式２と方式３の検索方式に従った２のｐ乗分木ノードデータ、および、次転送先アドレス、出力ルーティングプロセッサ２１０の番号、及び、出力ポート２４０の番号が格納されているとする。

【０１０３】なお、以下では方式２と方式３とを組合わせた例について説明するが、方式１と方式３とを組み合わせた場合でも同様なフローチャートに従うことで実現可能である。また、このフローチャートに従うことにより、ソフトウェアでもハードウェアでも経路検索処理を実現することができる。ソフトウェアで実現する場合、図４０の経路検索処理部２１３にはＣＰＵを用いればよい。またハードウェアで実現する場合、図４０の経路検索処理部２１３を専用ＬＳＩで構成すればよい。

【０１０４】図４１の処理８１０は木構造検索処理であり、処理８１１は経路情報出力処理である。まず、木構造検索処理８１０について説明する。

【０１０５】図４０の経路検索処理部２１３は、受信パケットの宛先ＩＰアドレスと初段ノードのマスク長ｍの値から初段ノードへのポインタを生成し、このポインタと、宛先ＩＰアドレスの第ｍビットから第（ｍ＋ｐ－１）ビットの値（以下、検査ビット値と呼ぶ）に従って経路テーブルメモリ２１４に格納されている初段ノードの読み込みアドレスを生成し、経路テーブルメモリ２１４から

該初段ノードの一部を読み込む(図４１の８００)。

【０１０６】次に、図４０の経路検索処理部２１３は、受信パケットの宛先ＩＰアドレスにノードのマスク長だけ上位ビットから有効とするマスクをかけたものと、ノードの経路アドレスとを比較し（図４１の８０１）、不一致の場合は木構造検索処理（図４１の８１０）を終了する(図４１の８０９)。一致する場合は図４１の処理８０２に進む。図４１の処理８０２、および８０３は、最長一致検索を実現するための経路情報の更新処理である。経路情報は、図３４乃至図３７に示すＦｌａｇ、次転送先アドレス、出力ルーティングプロセッサ（図３９の２１０）の番号、及び、出力ポート（図３９の２４０）の番号である。図４０の経路検索処理部２１３は、ノードデータの内のＦｌａｇ中のエントリ有りフラグを検査し、エントリ有りフラグの値が１の場合(図４１の８１２)のみ、読み込んだノード内の新たな経路情報をレジスタに保持する(図４１の８０３)。エントリ有りフラグの値が０の場合は更新処理を行わない(図４１の８１３)。

【０１０７】次に、図４０の経路検索処理部２１３は、図４０の経路テーブルメモリ２１４が出力するノードデータの内の次ノードへのポインタがＮＵＬＬかどうかを判定し、ＮＵＬＬの場合は木構造検索処理８１０を終了する。ＮＵＬＬでない場合はそのポインタと、新たな検査ビットの値に従い経路テーブルメモリ２１４に格納されている初段ノードの読み込みアドレスを生成し、経路テーブルメモリ２１４から該ノードデータを読み込む(図４１の８０５)。

【０１０８】上記の処理を繰り返すことにより、２のｐ乗分木方式の経路検索を行うことができる。

【０１０９】次に、図４１の経路情報出力処理８１１について説明する。木構造検索処理の結果、図４０の経路検索処理部２１３内に経路情報が保持されている。図４０の経路検索処理部２１３は、まず上記のエントリ有りフラグを調べ(図４１の８０６)、その値が０の場合は経路検索処理を終了し、転送処理部２１１へ検索結果無しという通知をする。エントリ有りフラグの値が１の場合は、検索の結果、あるエントリに一致したことになるため、経路情報を転送処理部２１１へ出力する。

【０１１０】次に、本発明の一実施例として、図４１で説明した検索方式をハードウェアで実現する場合の構成例を図４２、図４３を用いて説明する。

【０１１１】図４２に経路検索処理部２１３をハードウェアで構成した場合の構成例を示す。経路検索処理部２１３は、木構造検索回路２１３０と、読み込みアドレス生成回路２１３１と、経路検索処理制御回路２１３２とからなる。

【０１１２】木構造検索回路２１３０は、経路テーブルメモリ２１４に格納された２のｐ乗分木構造を検索し、次に読み込むべきノードのポインタの生成、受信パケットの宛先ＩＰアドレスの検査ビット値の抽出、木構造検索の終了判定、検索結果である経路情報の候補の更新を行う。また、読み込みアドレス生成回路２１３１は、木構造検索回路２１３０から出力される読み込むべきノードへのポインタ、および検査ビット値に従い、実際に読み込むノードの一部のワードのメモリアドレスを生成する。また、経路検索処理制御回路２１３２は、経路検索処理部２１３全体の制御(各回路の動作タイミングおよび動作状態管理など)を行う。

【０１１３】次に、経路検索処理部２１３の動作について図４２を用いて説明する。また、木構造検索回路２１３０の詳細動作については、図４３を用いて後述する。

【０１１４】木構造検索回路２１３０は、転送処理部２１１から受信パケットの宛先ＩＰアドレスを受け取り、この宛先ＩＰアドレスとノードのマスク長の値から次ノードへのポインタを生成して、読み込みアドレス生成回路２１３１に渡す。また、木構造検索回路２１３０は、ノードのマスク長で示される宛先ＩＰアドレスの検査ビット位置の値(検査ビット値)を抽出して、読み込みアドレス生成回路２１３１に渡す。

【０１１５】読み込みアドレス生成回路２１３１はこのノードへのポインタと、検査ビット値と、経路検索処理制御回路２１３２からのタイミング信号を用いて、読み出すべきノードデータが格納されているメモリアドレスを生成し、メモリ制御回路２１３２へ送信し、メモリ制御回路２１３２は上記メモリアドレスと経路検索処理制御回路２１３２からのタイミング信号を用いてメモリ制御信号を生成し、経路テーブルメモリ２１４へ転送する。上記のメモリ制御信号を受信した経路テーブルメモリ２１４は、対応するノードデータを信号線２１５を用いて木構造検索回路へ転送する。

【０１１６】木構造検索回路２１３０はこのノードデータを用いて、図４１の処理８０１、８０２、８０３、８０４、８０５を行う。これらの処理の詳細は図４３で後述する。図４１の処理８０１および８０４に対応する判定処理において木構造検索を終了すると判定した場合は、木構造検索終了信号を経路検索処理制御回路２１３２へ出力し、経路検索処理制御回路２１３２は、木構造検索回路２１３０内に保持された経路情報のうちのエントリ有りフラグを調べ、その値が０の場合は経路検索処理を終了し、転送処理部２１１へ検索結果無しという通知をする。エントリ有りフラグの値が１の場合は、経路情報を出力し検索処理を終了し、次のパケット処理の制御を行う。

【０１１７】次に、図４２の木構造検索回路２１３０の詳細を図４３を用いて説明する。

【０１１８】まず、図４１の初段ノードリード処理８００および次ノードリード処理８０５に対応する処理について、図４３を用いて説明する。転送処理部２１１より信号線２１６を用いて渡される宛先ＩＰアドレスは、宛

先ＩＰアドレスバッファ２１３００１内に保持され、初段ノードへのポインタ生成回路２１３００２、検査ビット抽出回路２１３００６、マスク処理回路２１３０１０に入力される。

【０１１９】初段ノードへのポインタ生成回路２１３００２は、予め図３９のルータ管理部２６０から初段ノードマスク長レジスタ２１３０１０に設定された値mに従い、受信パケットの経路アドレスの第０ビットから第(m－１)の上位mビットの値を抽出し、このmビットの値に従って初段ノードへのポインタを生成してアドレスセレクタ２１３００３に出力する。アドレスセレクタ２１３００３は、信号線２１３４によって図４２の経路検索処理制御回路２１３２から出力される初段ノード読み込み／初段ノード以外のノード読み込み選択信号に従い、初段ノードの読み込み時は、上記の初段ノードへのポインタを選択して図４２の読み込みアドレス生成回路２１３１に出力する。また、アドレスセレクタ２１３００３は、初段ノード以外のノードの読み込み時は、次ノードへのポインタバッファ２１３００４に保持されている次ノードへのポインタを選択して読み込みアドレス生成回路２１３１に出力する。

【０１２０】また、上記の処理と並行して、検査ビット抽出回路２１３００６は、初段ノード読み込み時には、検査ビット位置セレクタ２１３００５において選択されて出力される初段ノードマスク長レジスタ２１３０００の設定値mに従い、経路アドレスの第mビットから第(m＋ｐ－１)ビットまでのｐビットの検査ビット値を抽出して図４２の読み込みアドレス生成回路２１３１へ出力する。また、初段ノード以外のノードの読み込み時は、後述する次ノードマスク長バッファ（m１）２１３００７に保持されている次ノードのマスク長ｍ１が検査ビット位置セレクタ２１３００５において選択され、この次ノードのマスク長ｍ１が検査ビット抽出回路２１３００６に入力され、検査ビット抽出回路２１３００６は、この次ノードのマスク長ｍ１に従い、経路アドレスの第ｍ１ビットから第(ｍ１＋ｐ－１)ビットまでのｐビットの検査ビット値を抽出して図４２の読み込みアドレス生成回路２１３１へ出力する。

【０１２１】図４２の読み込みアドレス生成回路２１３１およびメモリ制御回路２１３３は、上記のノードへのポインタとｐビットの検査ビット値を用いて、図３６乃至図３７に示した順にノード内の各ワードのアドレスを生成し、経路テーブルメモリ２１４は入力されたメモリアドレスに格納されているノードのデータを信号線２１５に出力する。

【０１２２】信号線２１５のビット幅が３２ビットの場合、図３６で説明した方式を採用するとノードデータの一回の読み込みワード数は４ワードとなり、これらのデータは図３６の表に示した順番で信号線２１５に出力され、各バッファ(２１３００４、２１３００７、２１３０１３、２１３０１４、２１３０１５)に保持される。各バッファの保持タイミングは経路検索処理制御回路２１３２からの制御信号(図示していない)により制御される。バッファ２１３００４には次ノードへのポインタが保持され、バッファ２１３００７には次ノードのマスク長が保持され、バッファ２１３０１３にはノードのサブネットワークアドレスが保持され、バッファ２１３０１４にはフラグが保持される。

【０１２３】次に、図４１の経路アドレス一致比較処理８０１に対応する処理について、図４３を用いて説明する。初段ノード読み込み時には、マスク長セレクタ２１３００９を介して初段ノードマスク長レジスタ２１３０００の設定値がマスク処理回路２１３０１０に入力される。初段ノード以外のノード読み込み時には、マスク長セレクタ２１３００９を介してマスク長バッファ２１３００８に保持されているノードのマスク長が入力される。このマスク長バッファ２１３００８の値は、一つ前に読み込んだノードに格納されている次ノードのマスク長の値であり、次ノードマスク長バッファ２１３００７に保持されていたものである。次ノードマスク長バッファ２１３００７の値は、ノードの読み込み毎に更新されるため、更新される前に、現在読み込んでいるノード(以下、現ノードと呼ぶ)の経路アドレス一致比較処理に使用する現ノードのマスク長をマスク長バッファ２１３００８に保持しておく。

【０１２４】マスク処理回路２１３０１０は、これらのマスク長だけ上位ビットから有効とするマスクを生成し、このマスクと、宛先ＩＰアドレスバッファ２１３００１から出力される宛先ＩＰアドレスとの論理積をとり、その結果(以下、マスクした受信パケットの宛先ＩＰアドレスと呼ぶ)が一致比較回路２１３０１１に入力される。また、サブネットワークアドレスバッファ２１３０１３に保持されているノードのサブネットワークアドレスも一致比較回路２１３０１１に入力される。一致比較回路２１３０１１は、このノードのサブネットワークアドレスと、上記で説明した、マスクされた受信パケットの宛先ＩＰアドレスとを比較し、その結果が不一致となる場合に不一致信号を信号線２１３０１９を用いて木検索終了判定回路２１３０１２に出力する。木検索終了判定回路２１３０１２は上記不一致信号を入力し、木構造検索終了信号を図４２の経路検索処理制御回路２１３２に出力する。

【０１２５】また、マスク長バッファ２１３００８の値は、上記の経路アドレス比較が行われた後、次ノードマスク長バッファ２１３００７に保持されている次ノードのマスク長の値によって更新され、次のノードの読み込み時のサブネットワークアドレス比較に使用される。

【０１２６】次に、図４１の処理８０２、および８０３に対応する処理について、図４３を用いて説明する。

【０１２７】一致比較回路２１３０１１におけるサブネ

ットワークアドレス比較の結果、一致した場合、一致比較回路２１３０１１は信号線２１３０２０を用いて一致信号を更新判定回路２１３０１８に出力する。この一致信号が更新判定回路２１３０１８に入力され、かつ、フラグバッファ２１３０１４に保持されているフラグの内のエントリ有りフラグの値が１の場合(図４１の８１２)のみ、更新判定回路２１３０１８は更新信号をフラグ候補バッファ２１３０１６および経路情報候補バッファ２１３０１７に出力する。更新信号を受信したフラグ候補バッファ２１３０１６は、フラグバッファ２１３０１４に保持されているフラグを新たに保持し、同じく更新信号を受信した経路情報候補バッファ２１３０１７は、経路情報バッファ２１３０１５に保持されている経路情報を新たに保持する(図４１の経路情報更新処理８０３)。エントリ有りフラグの値が０の場合は更新判定回路２１３０１８は更新信号を送信しないので、フラグ候補バッファ２１３０１６および経路情報候補バッファ２１３０１７は更新処理を行わない(図４１の８１３)。

【０１２８】経路情報として上記のフラグ、及び、経路情報以外の情報を追加する必要がある場合は、木構造のノード内にそれらの情報を追加し、それらの情報を保持、更新するバッファを新たに追加すればよい。

【０１２９】次に、図４１の次ノードへのポインタがＮＵＬＬかどうかの判定処理８０４に対応する処理について、図４３を用いて説明する。次ノードへのポインタバッファ２１３００４内に保持されている次ノードへのポインタは木検索終了判定回路２１３０１２に入力される。この次ノードへのポインタがＮＵＬＬの場合、木検索終了判定回路２１３０１２は図４２の経路検索処理制御回路２１３２に木検索終了信号を出力する。

【０１３０】以上、図４２の木構造検索処理回路２１３０の動作を説明したが、ハードウェアで構成するため、図４１の木構造検索処理の中の８０１、８０２、８０３、８０４、８０５の各処理は逐次処理をする必要はなく、各処理に必要なデータが各バッファ２１３００４、２１３００７、２１３０１３、２１３０１４、２１３０１５に保持された後に、各処理を開始すればよく、上記の各処理を並列処理を行うことにより、高速に木構造の検索を行うことができる。

【０１３１】

【発明の効果】前述の説明の通り、本発明はルータ等のネットワーク中継装置に用いて好適なネットワーク次転送先検索方法及びそれを用いたネットワーク中継装置であり、ネットワーク中継装置が受信したパケットの転送先アドレスを高速に検索し、ネットワーク中継装置のパケット処理性能を向上させることができる。

【図面の簡単な説明】

【図１】本発明が前提とする一般的なネットワークシステムの構成図である。

【図２】ルータにおける経路検索仕様を説明する図である。

【図３】アドレス長３ビットの場合の全てのノードがある２分木を説明する図である。

【図４】アドレス長３ビットの場合の経路テーブル例を示す図である。

【図５】経路が割り付けられておらず、かつ、経路付きのノードへの途中経路にもなっていないノードを取り除いた木を説明する図である。

【図６】アドレス長３２ビットの場合の経路テーブル例を示す図である。

【図７】図６に示した経路テーブルに対応する木を説明する図である。

【図８】枝別れも経路の割り付けもないノードを取り除いた木を説明する図である。

【図９】図９はマスク長２ビットのノードをメモリ上に展開し第０～第１ビットの検索時間を除いた木を説明する図である。

【図１０】マスク長ｋビット目までのノードを経路検索ＬＳＩ内に入れた場合のメモリ構成図である。

【図１１】従来のパイプライン処理を行わない経路検索処理のタイムチャートを示す図である。

【図１２】マスク長ｋビット目までのノードを経路検索ＬＳＩ内に入れた場合の経路検索のパイプライン処理を表すタイムチャートを示す図である。

【図１３】２分木から８分木への変形時に一つの８分木ノードにまとめられる２分木ノードを囲んだ木を説明する図である。

【図１４】２分木から８分木への変形時に一つの８分木ノードにまとめられる２分木ノードを囲んだ木を説明する図である。

【図１５】２分木から８分木への変形時に一つの８分木ノードにまとめられる２分木ノードを囲んだ木を説明する図である。

【図１６】マスク長が０ビットから始まらないようにビット位置を区切った場合に最初のノードをメモリ上に展開することにより区切り位置までのビットの検索を行う木を説明する図である。

【図１７】マスク長が０ビットから始まらないようにビット位置を区切った場合に最初のノードをメモリ上に展開することにより区切り位置までのビットの検索を行う木を説明する図である。

【図１８】図９と組み合わせ、先頭のさらに多くのビット数の検索時間を除いた木を説明する図である。

【図１９】図１６と組み合わせ、先頭のさらに多くのビット数の検索時間を除いた木を説明する図である。

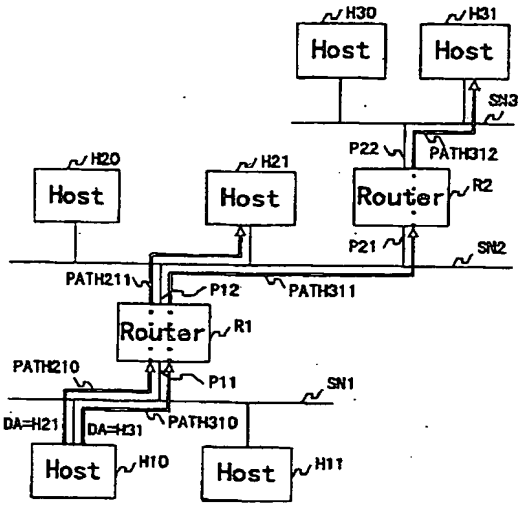
【図２０】図１７と組み合わせ、先頭のさらに多くのビット数の検索時間を除いた木を説明する図である。

【図２１】２分木から４分木への変形時に一つの４分木ノードにまとめられる３つの２分木ノードを２つの２分木ノード分につぶした木を説明する図である。

【図２２】一つの４分木ノードにまとめられる３つの２分木ノードの２つの２分木ノードへのつぶし方を説明する図である。
【図２３】一つの４分木ノードにまとめられる３つの２分木ノードの２つの２分木ノードへのつぶし方を説明する図である。
【図２４】一つの４分木ノードにまとめられる３つの２分木ノードの２つの２分木ノードへのつぶし方を説明する図である。
【図２５】一つの４分木ノードにまとめられる３つの２分木ノードの２つの２分木ノードへのつぶし方を説明する図である。
【図２６】一つの４分木ノードにまとめられる３つの２分木ノードの２つの２分木ノードへのつぶし方を説明する
【図２７】一つの４分木ノードにまとめられる３つの２分木ノードの２つの２分木ノードへのつぶし方を説明する図である。
【図２８】一つの４分木ノードにまとめられる３つの２分木ノードの２つの２分木ノードへのつぶし方を説明する図である。
【図２９】一つの４分木ノードにまとめられる３つの２分木ノードの２つの２分木ノードへのつぶし方を説明する図である。
【図３０】一つの４分木ノードにまとめられる３つの２分木ノードの２つの２分木ノードへのつぶし方を説明する図である。
【図３１】一つの８分木ノードにまとめられる７つの２分木ノードの４つの２分木ノードへのつぶし方を説明する図である。
【図３２】木の構成に必要なメモリ量見積もり時の木の広がりのモデルを説明する図である。
【図３３】木を２分木ノードで構成した場合と４、８分木ノードノードで構成した場合のノードの存在確率を考慮したメモリ量の比較を示す図である。
【図３４】２分木ノードの構造を示す図である。
【図３５】４分木ノードの構造を示す図である。
【図３６】図３６はノードの大きさが大きくなったときにノードデータリード時間の増大を防ぐために一つのノード全てを読み込まずに一部だけを読み込む方法を示す図である。
【図３７】条件によっては読み込む必要が無い要素を、条件に従い読み込まないようにすることで読み込みの時間を削減することで高速化を図る方法を示す図である。
【図３８】ルータ装置の一構成例を示すブロック構成図である。
【図３９】ルータ装置の他の構成例を示すブロック図である。
【図４０】本発明の一実施例であるルータ装置内ルーティングプロセッサのブロック図である。
【図４１】本発明の一実施例である経路検索処理部の検索処理フローチャートである。
【図４２】本発明の一実施例である経路検索処理部のブロック図
【図４３】本発明の一実施例である、木構造の経路検索を行う木構造検索処理回路のブロック図である。
【符号の説明】
１００、２００…ルータ、１１０…ルーティング制御部、１２０…ルータバス、１３０…ネットワークインタフェース部、１４０…ポート、１５０…サブネットワーク、２２０…ルータ装置内通信手段、２３０…ネットワークインタフェース部、２４０…ポート、２５０…サブネットワーク、２６０…ルータ管理部、２１１…転送処理部、２１２…パケットバッファメモリ、２１３…経路検索処理部、２１４…経路テーブルメモリ、２１５…経路テーブルメモリリードデータ、２１６…宛先ＩＰアドレス、８１０…木構造検索処理、８１１…経路情報出力処理、２１３０…木構造検索回路、２１３１…読み込みアドレス生成回路、２１３２…経路検索処理制御回路、２１３３…メモリ制御回路、２１３４…タイミング制御信号、２１３０００…初段ノードマスク長レジスタ、２１３００１…宛先ＩＰアドレスバッファ、２１３００２…初段ノードへのポインタ生成回路、２１３００３…アドレスセレクタ、２１３００４…次ノードへのポインタバッファ、２１３００５…検査ビット位置セレクタ、２１３００６…検査ビット抽出回路、２１３００７…次ノードマスク長バッファ（ｍ１）、２１３００８…マスク長バッファ、２１３００９…マスク長セレクタ、２１３０１０…マスク処理回路、２１３０１１…一致比較回路、２１３０１２…木検索終了判定回路、２１３０１３…サブネットワークアドレスバッファ、２１３０１４…フラグバッファ、２１３０１５…経路情報バッファ、２１３０１６…フラグ候補バッファ、２１３０１７…経路情報候補バッファ、２１３０１８…更新判定回路、２１３０１９…不一致出力信号、２１３０２０…一致出力信号。

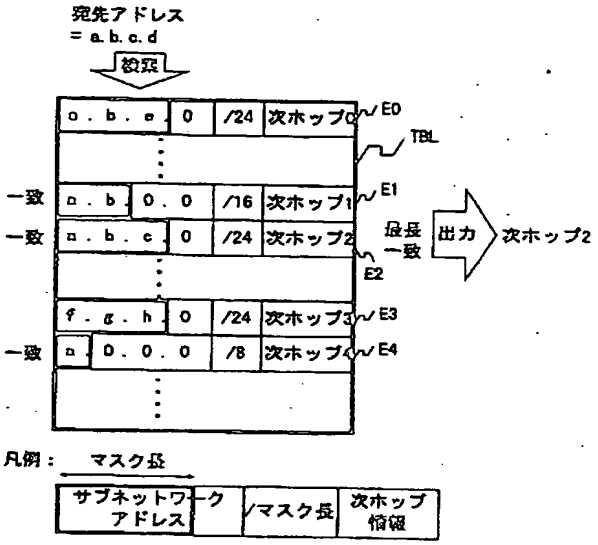
【図１】

図1



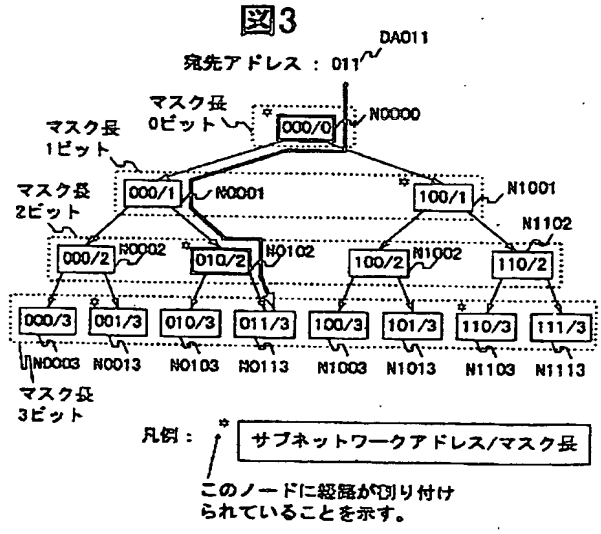
【図２】

図2



【図３】

図3



【図４】

図4

経路テーブル

サブネットワークアドレス	/	マスク長
000	/	0
001	/	3
010	/	2
100	/	1
110	/	3

【図６】

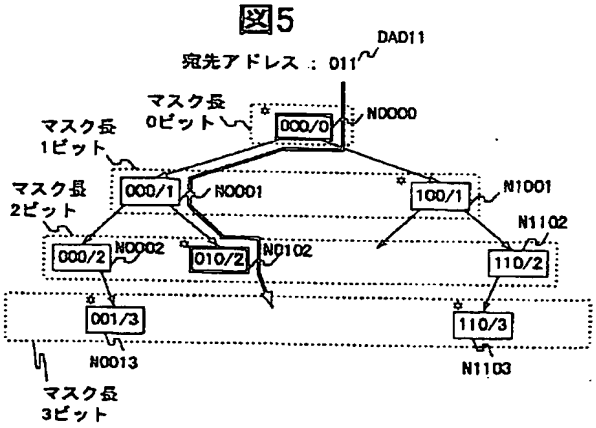
図6

経路テーブル

サブネットワークアドレス	/	マスク長
0x00. 00. 00. 00	/	0
0x85. 04. 00. 00	/	16
0x85. 05. 00. 00	/	16

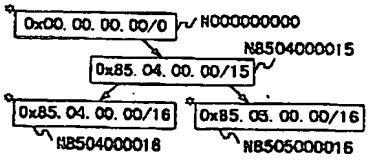
【図５】

図5



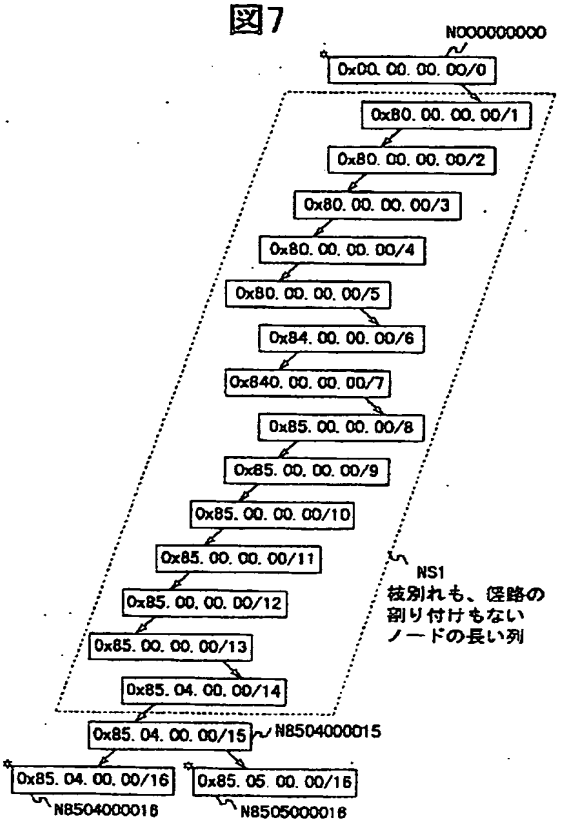
【図８】

図8



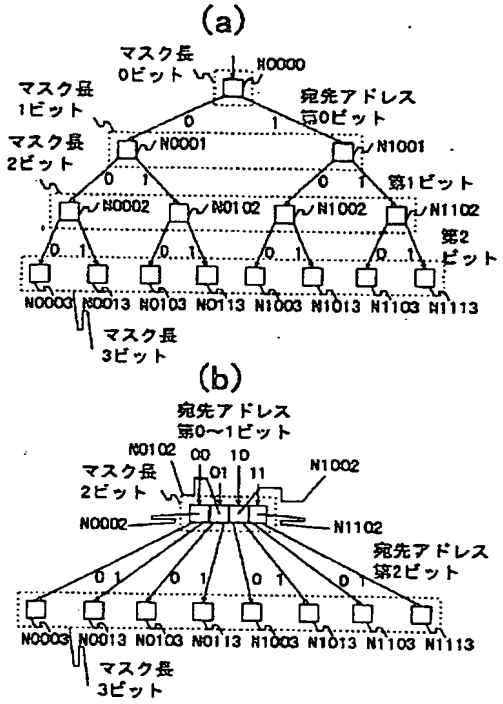
【図７】

図7



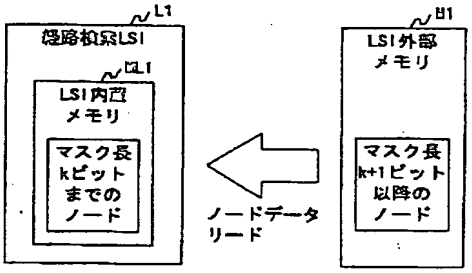
【図９】

図9



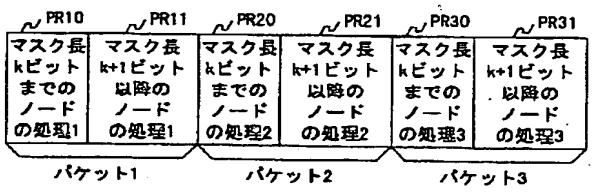
【図１０】

図10



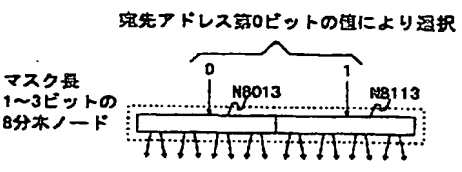
【図１１】

図11



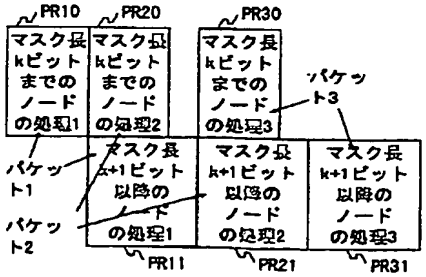
【図１６】

図16



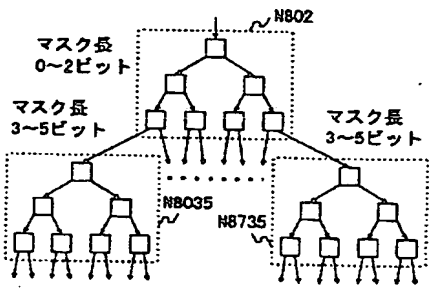
【図１２】

図12



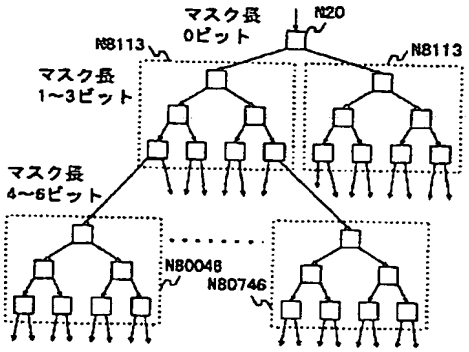
【図１３】

図13



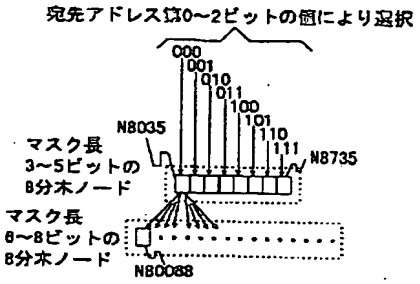
【図１４】

図14



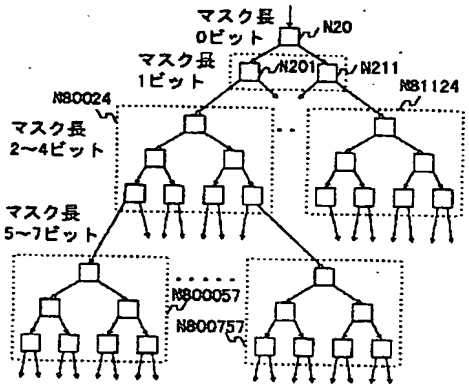
【図１８】

図18



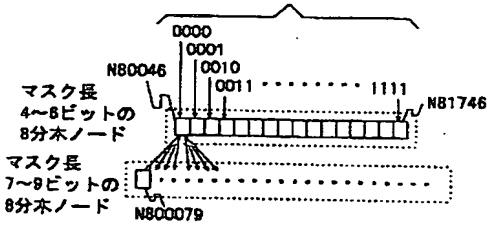
【図１５】

図15



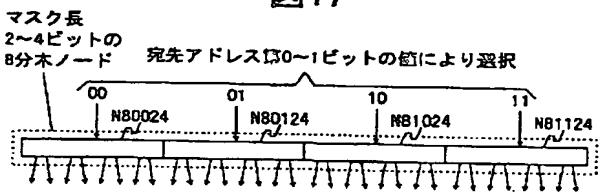
【図１９】

図19



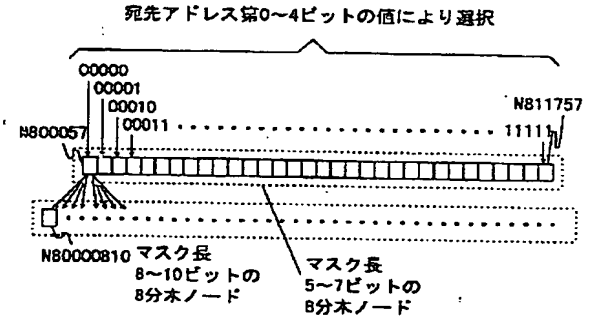
【図１７】

図17



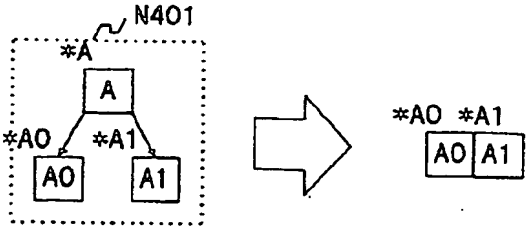
【図２０】

図20



【図２２】

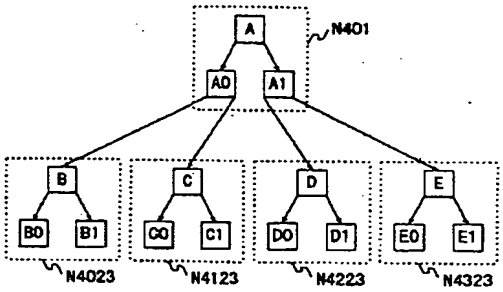
図22



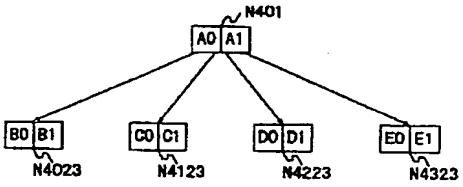
【図２１】

図21

(a)

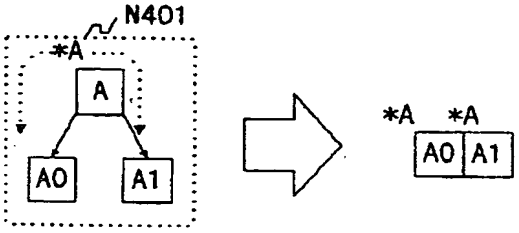


(b)



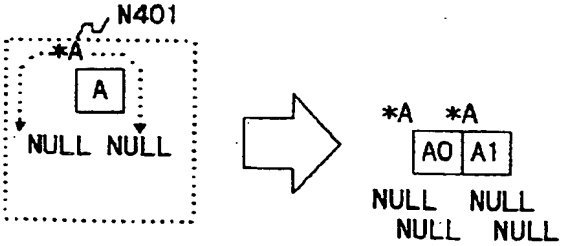
【図２４】

図24



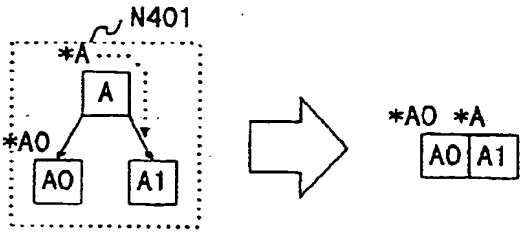
【図２６】

図26



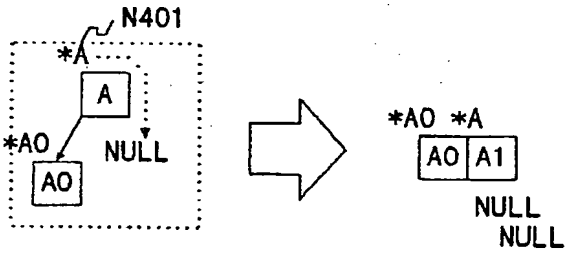
【図２３】

図23



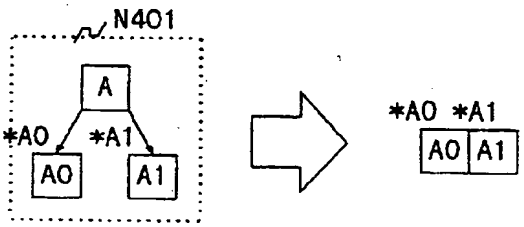
【図２５】

図25



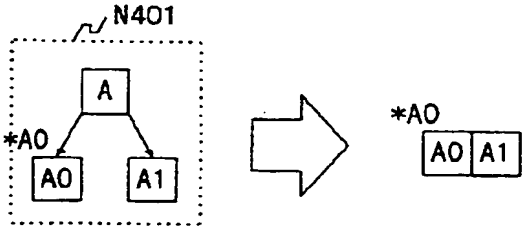
【図２７】

図27



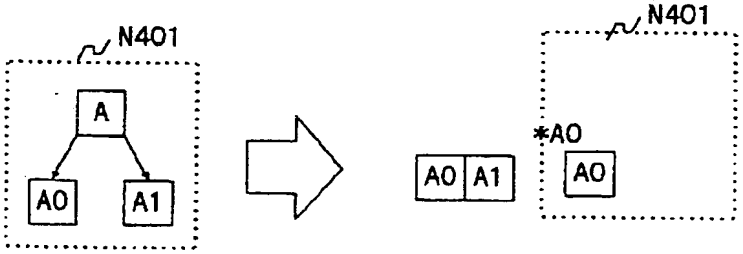
【図２８】

図28



【図２９】

図29



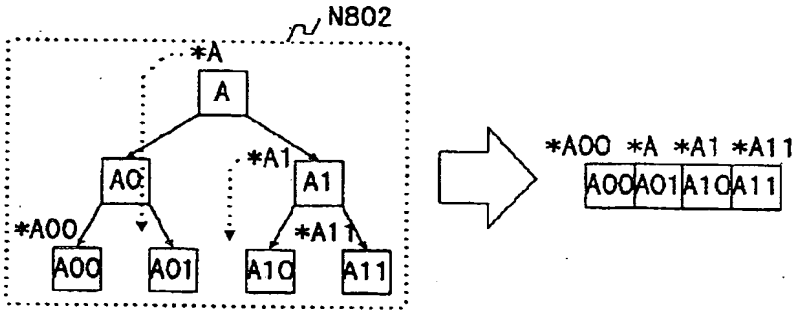
【図３０】

図30

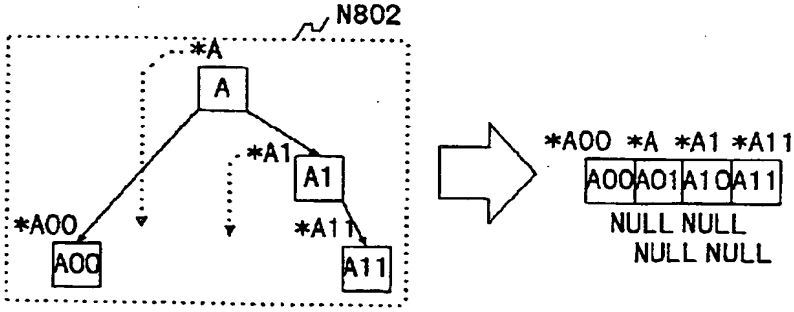
【図３１】

図31

(a)

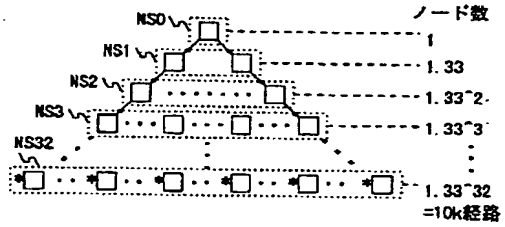


(b)



【図３２】

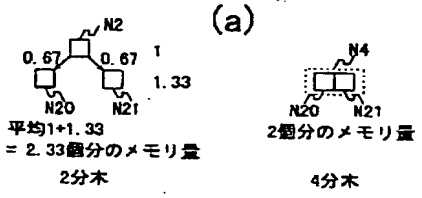
図32



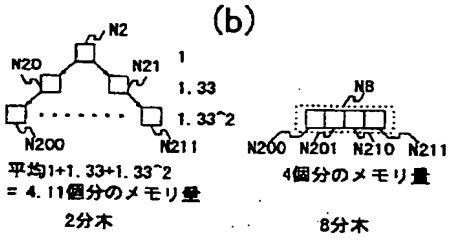
【図３３】

図33

(a)

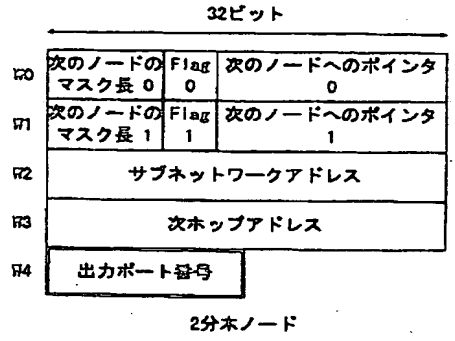


(b)



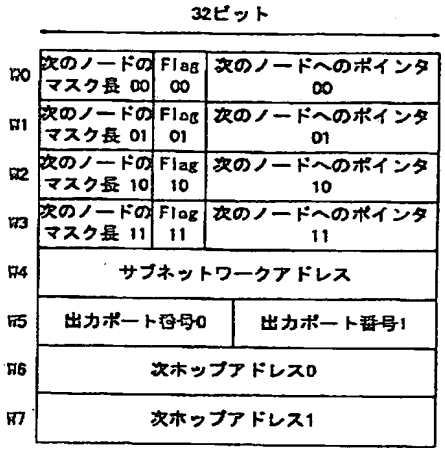
【図３４】

図34



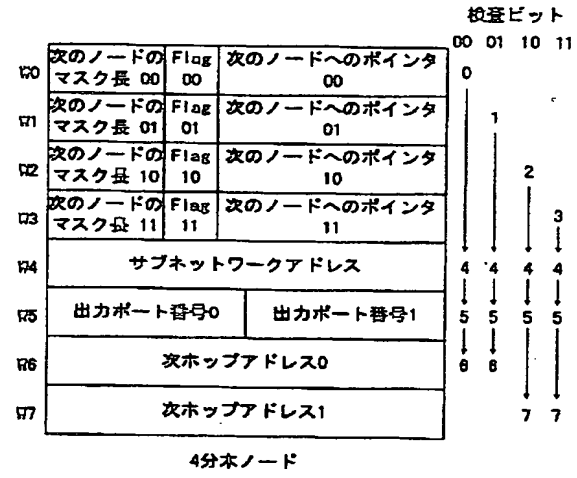
【図３５】

図35



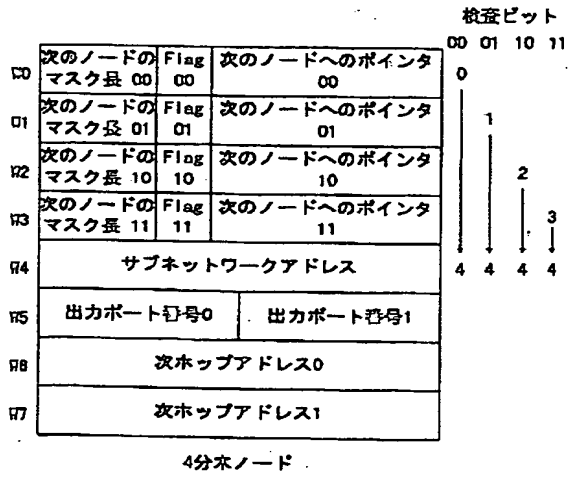
【図３６】

図36



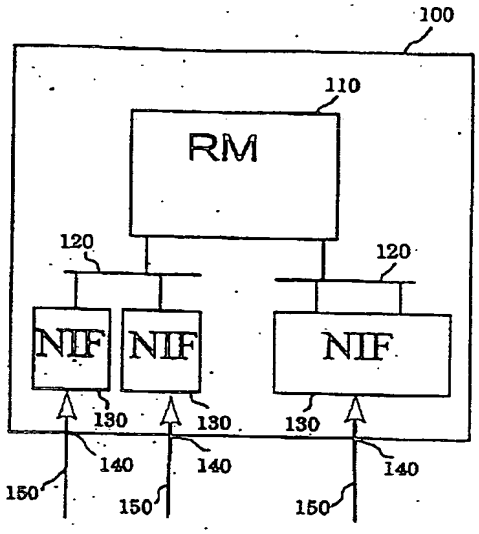
【図３７】

図37



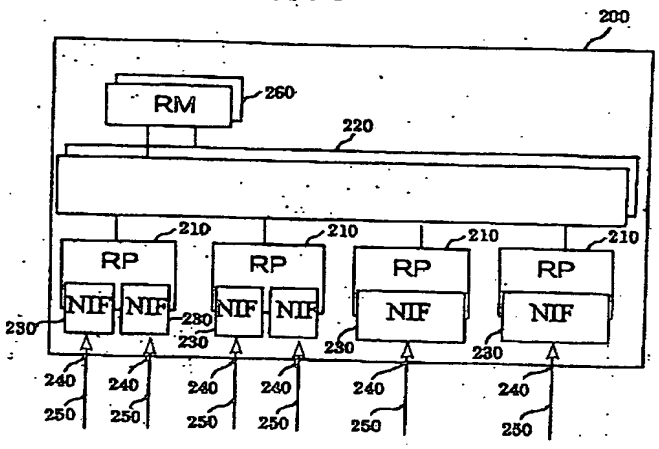
【図３８】

図３８



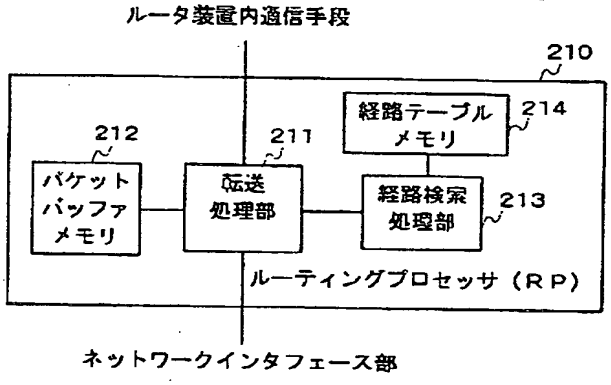
【図３９】

図３９



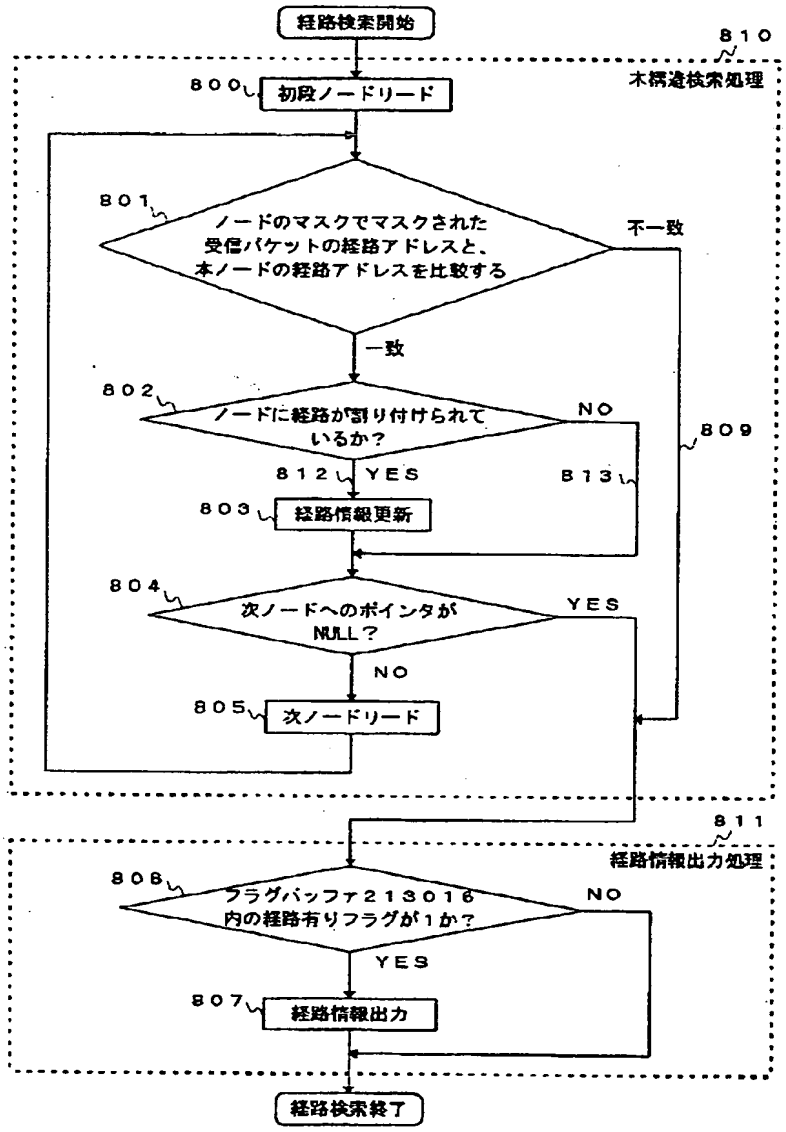
【図４０】

【図４０】



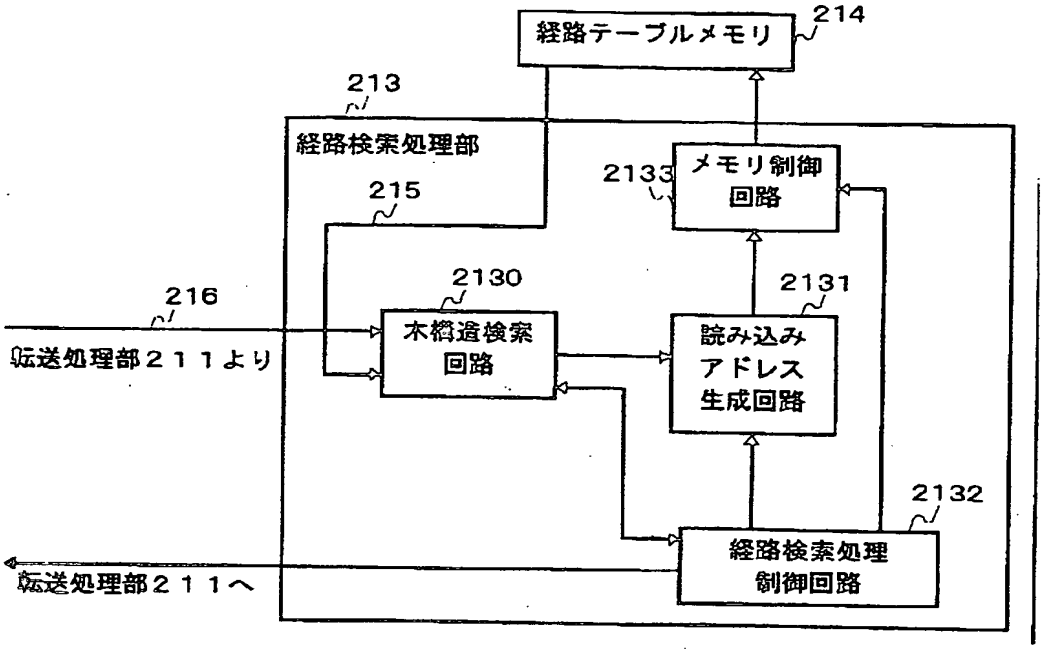
【図４１】

【図４１】


経路検索開始
810
木構造検索処理
800
初段ノードリード
801
ノードのマスクでマスクされた
受信パケットの経路アドレスと、
本ノードの経路アドレスを比較する
不一致
一致
802
ノードに経路が割り付けられて
いるか？
NO
809
812
YES
813
803
経路情報更新
804
次ノードへのポインタが
NULL？
YES
NO
805
次ノードリード
811
経路情報出力処理
808
フラグバッファ２１３０１６
内の経路有りフラグが１か？
NO
YES
807
経路情報出力
経路検索終了

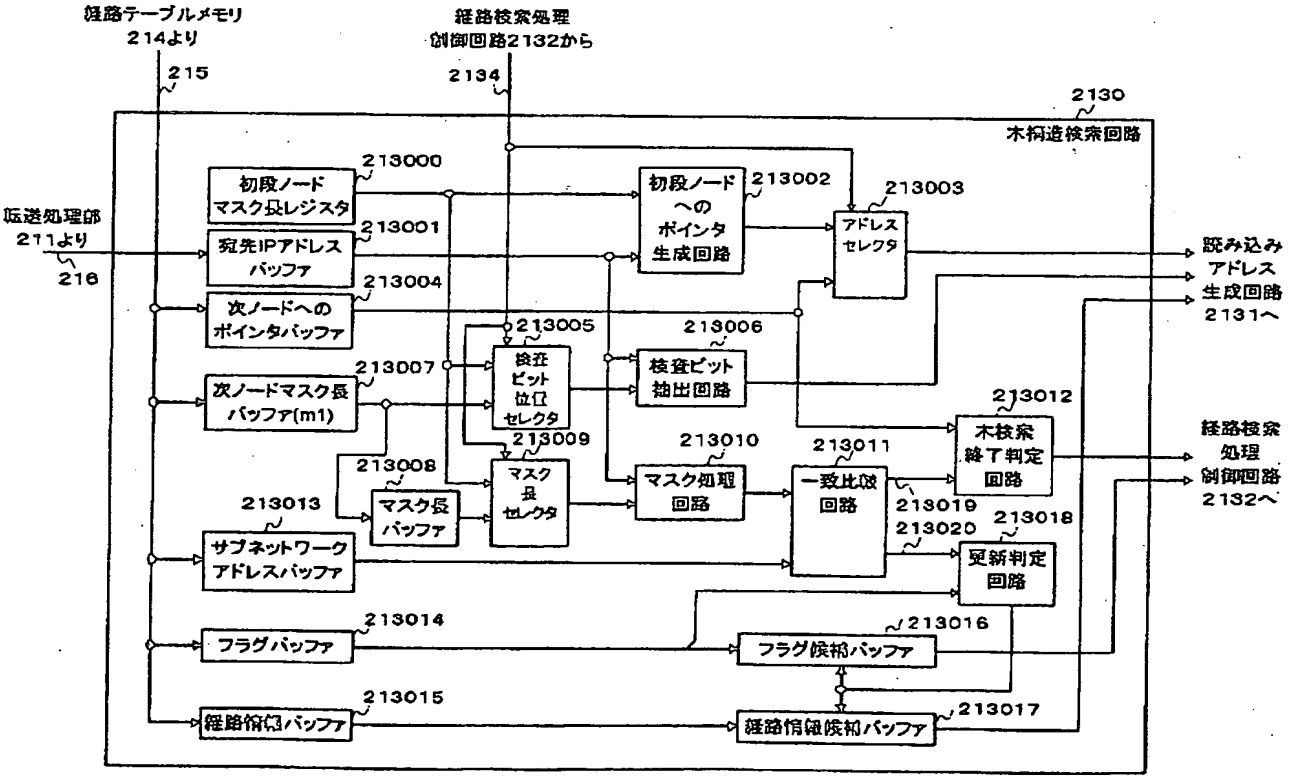
【図４２】

【図４２】



【図４３】

【図４３】



フロントページの続き

(72)発明者 赤羽 真一
東京都国分寺市東恋ヶ窪一丁目280番地
株式会社日立製作所中央研究所内

(72)発明者 左古 義人
神奈川県秦野市堀山下１番地 株式会社日
立製作所汎用コンピュータ事業部内

(72)発明者 田那邊 昇
神奈川県秦野市堀山下１番地 株式会社日
立製作所汎用コンピュータ事業部内